Virtual character: hybrid controller coupling multi-objective dynamic control and captured motions

Mingxing Liu

THÈSE DE DOCTORAT

DE L'UNIVERSITÉ PIERRE ET MARIE CURIE

École Doctorale de Science Mécanique, Acoustique, Électronique et Robotique de Paris

Spécialité

Mécanique - Robotique

Pour obtenir le grade de

DOCTEUR de l'UNIVERSITÉ PIERRE ET MARIE CURIE

Présentée par

Mingxing LIU

Personnage virtuel : contrôleur hybride couplant commande dynamique multi-objectifs et mouvements capturés

---

Soutenance prévue le 28 Septembre 2012

Devant le jury composé de :

M.	A. Kheddar	Directeur de Recherche au CNRS	Rapporteur
M.	M. Buss	Directeur de l'«Institute of Automatic Control Engineering », TUM	Rapporteur
M.	P. Fraisse	Professeur à l'Université de Montpellier 2	Examinateur
M.	V. Padois	Maître de Conférence à l'Université Pierre et Marie Curie	Examinateur
M.	F. Ben Amar	Professeur à l'Université Pierre et Marie Curie	Examinateur
M.	H. Arabi	Ergonome, PSA Peugeot Citroën, Vélizy	Examinateur
M.	P. Evrard	Chercheur au CEA LIST, Fontenay aux Roses	Examinateur
M.	A. Micaelli	Directeur de recherche au CEA LIST, Fontenay-aux-Roses	Directeur de Thèse

A dissertation submitted for the degree of

Doctor of Philosophy

of the Pierre and Marie Curie University

in

Mechanics - Robotics

Presented par

Mingxing Liu

Virtual character: hybrid controller coupling multi-objective dynamic control and captured motions

---

To be defended on September 28th, 2012

Committee in charge

M.	A. Kheddar	Research Director at CNRS	Referee
M.	M. Buss	Director of the Institute of Automatic Control Engineering , TUM	Referee
M.	P. Fraisse	Professor at University of Montpellier 2	Examiner
M.	V. Padois	Associate professor at the Pierre and Marie Curie University	Examiner
M.	F. Ben Amar	Professor at the Pierre and Marie Curie University	Examiner
M.	H. Arabi	Ergonomist, PSA Peugeot Citroën, Vélizy	Examiner
M.	P. Evrard	Researcher at CEA LIST, Fontenay aux Roses	Examiner
M.	A. Micaelli	Research director at CEA LIST, Fontenay-aux-Roses	Adviser

Résumé

Un grand défi pour les personnages virtuels est de pouvoir interagir avec des opérateurs humains en effectuant des tâches dans des environnements virtuels physiquement réalistes.

Cette thèse s'intéresse particulièrement à l'interaction avec des opérateurs faiblement immergés, c'est-à-dire avec des opérateurs disposant du minimum d'équipement nécessaire à l'interaction, par exemple, une simple capture de mouvement des mains. Cela implique de doter le personnage virtuel de la capacité d'ajuster ses postures de manière autonome, d'accomplir les tâches requises par l'opérateur en temps réel en tâchant de suivre au mieux ses mouvements, tout en gérant de manière autonome les multiples contraintes dues aux interactions avec l'environnement virtuel.

Il existe actuellement deux types de méthodes pour animer des personnages virtuels. La première issue du domaine *computer graphics* consiste à restituer une trajectoire préalablement enregistrée avec un système de capture de mouvements. La seconde proche du domaine de la robotique, consiste à commander les actionneurs du modèle dynamique associé au personnage virtuel afin d'atteindre plusieurs objectifs et en respectant des contraintes relatives à l'environnement. L'objectif de cette thèse est de trouver le meilleur couplage de ces deux méthodes afin de bénéficier de leurs avantages.

Cette thèse présente un système de contrôle hybride original qui permet de réaliser un personnage virtuel interactif avec certains niveaux de l'autonomie. Une approche *d'optimisation de posture* est proposée, qui permet au personnage virtuel de chercher des postures optimales et robustes, y compris les positions de contact, avant d'effectuer une tâche de manipulation donnée. Cette approche peut être utilisée soit dans la préparation d'une tâche de manipulation, soit pour l'évaluation de la faisabilité d'une tâche. Un cadre de *contrôle multi-objectif* est développé, pouvant gérer plusieurs objectifs de tâches et de multiples contacts unilatéraux et bilatéraux avec ou sans frottements. Il permet au personnage d'effectuer les tâches de suivi de mouvement et les tâches de manipulation d'objets dans un environnement virtuel physiquement réaliste, tout en interagissant avec un opérateur en temps réel. Un élément important développé dans ce cadre est une méthode de type *wrench-bound*. Cette méthode peut gérer les conflits des tâches et d'assurer la performance du contrôleur de la tâche de haute priorité. Il s'agit d'une nouvelle approche

de contrôle hiérarchisé comportant différents niveaux de priorité, permettant d'imposer des contraintes d'inégalité sur la tâche de haute priorité, tout en assurant la passivité du système pour garantir la stabilité des opérations.

Notre système de contrôle basé sur toutes les approches développées dans cette thèse permet à un personnage virtuel d'effectuer une variété de tâches, telles que le suivi des mouvements capturés et la manipulation d'objets. Il améliore les performances des tâches par la capacité du personnage virtuel de gérer les efforts d'interaction. Il permet également au personnage virtuel d'effectuer des séquences de tâches de manière autonome, afin d'atteindre un objectif à long terme.

Mots Clés : *Personnage virtuel, Contrôle basé sur la physique, Contrôle global du mouvement, Interaction humain-robot, Commande multi-objectifs, Contrôle hiérarchisé, Optimisation de posture.*

Abstract

A great challenge for interactive virtual characters is to be able to interact with human operators by performing tasks in physics-based virtual environments.

This dissertation is particularly interested in the interaction with operators who are slightly immersed, that is to say, with operators having the minimum equipment necessary for the interaction, for example, a simple motion capture of the hands. This involves endowing the virtual character with the ability to adjust its postures autonomously, and to accomplish tasks required by operators in real time by trying to follow their motions as best as it can, while autonomously handling multiple constraints due to interactions with virtual environments.

There are currently two types of methods to animate virtual characters. The first is related to the domain of *computer graphics*, consisting in restoring a captured motion sequence. The second is close to the domain of *robotics*, consisting in controlling the actuators of the dynamic model associated with a robot to achieve several objectives while respecting the constraints related to the environment. Objective of this dissertation is to find the best combination of these two types of methods, so as to take advantages of their benefits.

In this dissertation, we present an original hybrid control system that allows us to realize an interactive virtual character with certain levels of autonomy. We propose a *posture optimization approach*, which allows the virtual character to search for optimal and robust postures, including contact positions, before performing a given manipulation task. This approach can be used in either the preparation for a manipulation task, or the evaluation of the feasibility of a task. We develop a *multi-objective control framework*, which can handle multiple task objectives and multiple unilateral and bilateral contacts with or without friction. It allows the character to perform motion tracking and object manipulation tasks in a physics-based virtual environment, while interacting with an operator in real-time. An important component developed in this control framework is a *method to compute wrench bounds for prioritized control*, which can handle task conflicts and ensure the controller performance of the higher-priority task. It is a novel prioritized control, which allows inequality constraints on the higher-priority task, and can ensure

the passivity of the system to guarantee stable operations.

Our control system based on all the approaches developed in this dissertation can allow a virtual character to perform a variety of tasks, such as tracking captured motions and object manipulation. It can improve task performance by enhancing the character's ability to handle interaction forces. Moreover, it also allows the character to autonomously perform sequences of tasks to achieve a long-term goal.

Keywords: *Virtual character, Physics-based control, Whole-body motion control, Human-robot interaction, Multi-objective control, Prioritized control, Posture optimization.*

Acknowledgements

After the three years of my Ph.D. life, I feel happy that I have made a right decision three years ago to start this research work here. Not only because I have learned many things concerning my research topics, the way to work, and the life, but also because I really enjoyed the life experience with all the people around me. There are so many people that I am grateful to, and I hope I have not forgotten anyone in these concentrated acknowledgments.

First, I would like to thank Alain Micaelli, who has always been a very kind Ph.D. supervisor and a great source of inspiration for my work. I was so lucky to have the guidance and support of such a brilliant expert on automatic control and virtual characters.

Thanks to my committee members: Abderrahmane kheddar, Martin Buss, Philippe Fraisse, Vincent Padois, Faiz Ben Amar, Hossein Arabi, and Paul Evrard, for their time and comments.

This dissertation work is deeply attached to the Interactive Simulation Laboratory of CEA-LIST. A lot of thanks is due to all the lab members. Thanks to Paul Evrard and Adrien Escande, for their helpful collaborations and inspiring discussions on robotics and programming, as well as their insightful comments on my research work. I am grateful to Laurent Chodorge, who runs the lab, for supporting my Ph.D. work and for approving my participation in each conference and course. I would like to thank Claude Andriot for recommending related papers and works to us from time to time. I would also like to thank Julien Brisset and Boris Moriniere for helping me use the virtual reality room. Thanks to Mathieu Gautier, Jean Sreng, Geremie Le Garrec, Florian Bergez for helping me use the physics simulation software environment developed by the lab. Thanks to Xavier Merlhiot for his lessons of kinematics and dynamics when I just started in robotics. Thanks to Renaud Deligny, Romain Michalec, Bruno Bodin, Lancelot Perrotte, Philippe Gravez, Christophe Leroy, Guillaume Saupin, Schramm Florian, Anthony Truchet for all kinds of their help. Special thanks to Emmanuel Dhooge for being a wonderful teacher of French language and culture. Special thanks to Mathieu Gautier and Adrien Escande for lending me your French novels and comic books. I will never forget Flavia Desiro, Yi-ching Tsai, Zhaopeng Qiu, Giovanni de Magistris, Joris Vaillant, Olivier Roussel, Darine

Mansour, and Kevin Wargez, for all the good times that we have spent in the lab and in the restaurant.

Thanks to Christelle Quernee, Elodie Renividaud, Annick Latare, and Anne Goue at CEA, as well as to Anne-Marie Aubin, the manager of our doctoral school at UPMC, for their help on administrative issues.

Finally, I would like to thank my family. Thanks to my husband Hao Li for being with me through this challenging journey. Thanks to my lovely baby in belly for not making me feel any uncomfortable during the last months of my Ph.D. life. Most importantly, thanks to my parents Chengyun Liu and Chenhui Liu for their everlasting love, for their encourage and advice at every step of the way, and for their care wherever I am.

Contents

List of Figures

List of Tables

Nomenclature

$\dot{\mathbf{T}}$	acceleration in generalized coordinates
$\mathbf{E}$	kinetic energy
$\mathbf{F}$	force in Cartesian space
$\mathbf{H}$	4×4 homogeneous transformation matrix
$\mathbf{I}$	identity matrix
$\mathbf{J}$	Jacobian matrix
$\mathbf{L}$	matrix to select the actuated degrees of freedom ($\mathbf{L} = \begin{bmatrix} \mathbf{0}\ \mathbf{I} \end{bmatrix}^T$)
$\mathbf{M}$	generalized inertia matrix
$\mathbf{NT}$	centrifugal and Coriolis forces
$\mathbf{q}$	joint angles
$\mathbf{R}$	rotation matrix
$\mathbf{T}$	velocity in generalized coordinates
$\mathbf{U}$	potential energy
$\mathbf{V}$	twist ($\mathbf{V} = [\mathbf{v}^T, \boldsymbol{\omega}^T]^T$)
$\mathbf{v}$	linear velocity
$\mathbf{W}$	wrench
$\mathbf{X}$	position vector
$\boldsymbol{\Gamma}$	moment in Cartesian space
$\boldsymbol{\gamma}$	gravity force in generalized coordinates
$\boldsymbol{\omega}$	angular velocity
$\boldsymbol{\tau}$	the set of joint torques

$diag(a_1, ..., a_n)$ a diagonal matrix whose diagonal entries starting in the upper left corner are $a_1, ..., a_n$

$tr()$ matrix trace

Chapter 1

Introduction

Contents

Virtual characters are simulated digital characters that behave in interactive graphical virtual environments. They may come in all shapes and sizes. Virtual characters concerned in this dissertation can be on one hand as avatars, graphical representations of human bodies; and on the other hand as digital robots driven by robotic control systems. The goal of this dissertation is to develop approaches for the control of virtual characters that perform actions in virtual environments and interact with human operators.

1.1 Motivation

With the development of computer science and computerized control, virtual characters have come on the scene in computer games and virtual reality. Control issues in virtual characters can help us make computer animations more realistic. Moreover, the researches on virtual character control can help people to better understand the biomechanics of a human body, as well as how humans interact with their environments.

Today, virtual characters have also walked their way into industry. In industry design, interactions between human operator and a product can be modeled, ergonomics can be studied to improve human well-being, and overall system performance can be optimized.

Another application is virtual training environments, which are tools that allow a user to interact virtually with a working environment by animating his avatar. Such training environments are created to reduce training time and expenses. Training in virtual environments provides a tolerance for errors, which may not be acceptable during real practices. On the other hand, training environments also help to ensure the security of learners. Learners can first practice skills in virtual environments, before directly being exposed to such environments, which can be dangerous or difficult to realize in reality.

This dissertation aims to advance the state of the art in both robotics and computer graphics. We focus on the design of controllers that improve the behaviors of virtual characters. The control approaches developed here are also expected to be applied to real robots. This dissertation is also dedicated to enhance virtual characters' ability of interacting with human operators, by executing their orders with a high fidelity in virtual environments. Another motivation is to produce realistic computer animations, where virtual characters can handle interaction forces when performing tasks such as manipulating different objects.

1.2 Problem statement

This dissertation proposes a hybrid control system for interactive virtual characters, which are controlled by both human operators and a robotic controller.

The virtual character used in this dissertation is shown in Fig.1.1. Its body weighs $79kg$, with a height of $1.7m$. Its skeleton model can be considered as a set of rigid links connected with joints. Its body consists of 45 degrees of freedom (DoF), including 6 DoFs for the root position and orientation, and 39 DoFs for the joints. The joint DoFs are listed in Table 1.1. There are four contact points on each foot.

The system of an interactive virtual character can be decomposed into several hierarchical layers as described in Fig.1.2.

- The **Data Layer** collects data from all kinds of communication interfaces, such as data coming from motion capture systems, keyboards, space mouses, and so on. It can be used as an interface for receiving information during interactions with human operators.

Figure 1.1: The virtual human.

- The **Behavior Layer** decides *what to do* according to the information received from Data Layer. It is responsible for the behaviors to perform. Here a **behavior** refers to a sequence of actions that create responses to user inputs or character's intentions. A behavior may involve movements of the whole or a part of the character's body. A complex behavior can be decomposed into a sequence of other simpler behaviors. For example, a behavior of "moving an object" can be described as: walk towards an object if it is out of reach, grasp it, then (walk to) move it to a target position, and finally release it.
- The **Task Control Layer** solves the problem of *how to do*. This layer operates at a higher speed than the Behavior Layer. A task controller is implemented at this level, which computes joint torques at each simulation time step. The joint torques are used to drive the virtual character to accomplish the desired behaviors. Here a **task** means that a certain frame on the virtual human's body should move from its current state to a desired state. So a behavior can be decomposed into several low-level tasks. Taking the "moving an object" behavior for example, we can define a foot task for walking, a center of mass task for balance control, a hand task for hand motion control and object manipulation control, and a joint task for posture

Table 1.1: Degrees of freedom for the joints

Joint	Rotational DoFs
neck	3
torso	3
lumbar	3
hip (×2)	3 (×2)
knee (×2)	2 (×2)
ankle (×2)	2 (×2)
toe (×2)	1 (×2)
shoulder (×2)	3 (×2)
elbow (×2)	2 (×2)
wrist (×2)	2 (×2)

control.

- The **Physics Layer** takes as inputs joint torques, as well as external forces applied on the virtual character, then computes the kinematic states according to physical laws.
- The **Graphics Layer** outputs the visual appearance of the virtual character, such as its global position and posture.

The problems studied in this dissertation are related to the top three layers, especially the Task Control Layer. The virtual character is in real-time interaction with an operator through a motion capture system. It should figure out what actions to perform in order to achieve a global goal, based on interactions with the operator and the simulated events. Finally it should find appropriate joint torques to generate desired motions and fulfill desired tasks.

This dissertation focuses on the two main capabilities of virtual characters:

- The capability of interacting with a human operator in real-time. The characters are expected to be able to fulfill the tasks required by the operator.
- The capability of autonomously improving task performance. The characters are expected to autonomously adjust their motions for dealing with multiple contact

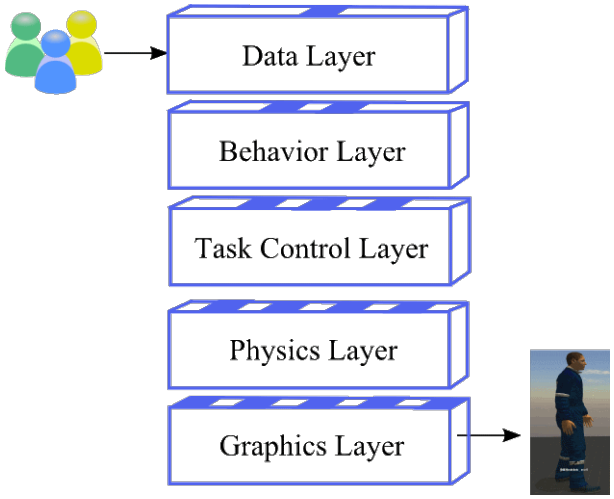


Figure 1.2: The system of a interactive virtual character.

forces and constraints due to interactions with virtual environments.

The main difficulty in our research topic is due to the fact that the character has to handle interactions with virtual environments. Since a virtual character's body is under-actuated, it should make contacts with the environment and use contact forces to perform desired movements. The operator's motions may be used as reference motions for virtual characters, but they can be inappropriate for a given task. This can be because

- the human operator is not completely immersed into the virtual environment and cannot sense forces applied to the character;
- or the operator is not able to perform the desired actions in his world, which is different from the virtual environment;
- or the operator is not skillful enough for providing suitable reference postures for some difficult tasks.

So the control system should be able to adjust the character's motions to adapt them to unknown external contact forces; so that the character can ensure the balance and fulfill the required tasks. To achieve this, the controller must take into account multiple objectives and constraints. Moreover, since a virtual character's body has many degrees of freedom (DoF), the controller must handle redundancy and find an optimal posture among all the possibilities.

1.3 A review of character control

One great concern in virtual character control is how to generate realistic interactive motions. There are currently two basic ways to animate virtual characters. The first one consists in restoring captured motion sequences. In this way, virtual characters are completely controlled by human operators by simply emulating their movements. The second one consists in controlling the actuators of the dynamic model associated with a robot to achieve several objectives and constraints. Such methods endow characters with a certain level of intelligence, allowing them to make decisions by themselves on how to perform tasks and react to environments, without constant guidance of human operators. This dissertation will show that, in order to improve task performance in virtual environments, virtual characters should have some autonomy, instead of being completely controlled by human operators.

1.3.1 Character animation using motion capture data

One aspect of this dissertation is the research on virtual character animation using motion capture techniques. Motion capture, or mocap, refers to techniques of recording movements of human bodies or other objects for analysis or playback. We have developed a control framework that allows virtual characters to track captured key frame motions of human operators in real-time. Related work on character control using motion capture data is briefly reviewed here.

Motion capture techniques have been widely used to restore a captured motion, so as to obtain natural and life-like motions. The idea of animating characters by using motion capture data can be traced back to the late 1970's, when a method called rotoscoping was used in animated films. A traditional way to realize such applications is to attach some optical trackers on an operator's body; then the positions of these markers are measured and recorded for driving a character. Such kind of application has first been realized in the context of computer graphics by the Graphical Marionette [Ginsberg] in 1983.

Today, a lot of work has been dedicated to the animation of virtual characters or robots based on motion capture data. Motion retargetting method has been used to adapt the captured motion from one character to other characters with different sizes or morpholo-

gies [Gleicher 1998,Monzani 2000]. In the domain of robotics, the approach of Direct Marker Control [Ott 2008,Lee 2010] has been applied for imitating human motions. The measured trackers on the operator's body are connected with corresponding points on the character's body by virtual springs, the forces generated by which help to guide the character's motions. Many motion synthesis techniques have been developed to generate new animations from a motion capture database. Some motion editing techniques, such as [Wiley 1997,Zordan 2005,Cooper 2007], use blending methods to produce new animations by interpolating motion capture data. Motion blending was combined with optimal control, which takes into account some kinematic constraints, to change captured motions for dynamic interactions with objects [Jain 2009a]. Inverse kinematics was combined with dynamic correction to modifies motion capture data to respond to physical collision forces during specified tasks [Zordan 2002,Multon 2009]. Reference locomotion motions have been adapted to new physically simulated environments by applying constrained inverse kinematics [Komura 2004], model predictive control [Da Silva 2008a], linear quadratic regulator [da Silva 2008b] and nonlinear quadratic regulator [Muico 2009].

The motion tracking that is presented in this dissertation shares the same objective with many motion synthesis approaches, which is to produce realistic physics-based motions from captured motions. The control approaches that have been developed aims at dynamically adapting captured motions to unpredictable simulated events, and trying to follow human operator's orders as close as possible.

1.3.2 Character control in physics-based environments

In real world and in physically simulated environments, motions should be generated subject to the system geometrics and dynamics constraints to ensure physical correctness. Constraints related to character's body, such as joint limits, should be respected. Since (virtual) humanoid characters are under-actuated systems, they should handle contact forces to realize desired physical interactions and maintain balance at the same time. One important problem with characters totally driven by human operators is that they may fail to fulfill a task due to poor human guidance, because human operators may not possess all the necessary information and conditions for guiding the character, as mentioned in Section 1.2. Therefore, this dissertation aims to endow characters with certain levels of autonomy

to make some decisions by themselves on how to handle objectives and constraints in physics-based environments.

Since a character is a complex system with many DoFs, a great number of posture configurations are possible. In order to simplify some high dimensional control problems, stability criteria and simplified models that abstract the most important characteristics of the complete model have been proposed. In many balance controllers, stability criteria such as the Center of Mass (CoM) and the Zero Moment Point (ZMP) have been used for the analysis of equilibrium. An algorithm was presented in [Bretl 2008] to compute the set of CoM positions (the support region), which ensures the static equilibrium of a legged robot making multiple non-coplanar contacts with any type of terrain. The work in [Harada 2004] formulated the region of ZMP that ensures the dynamic balance of a robot grasping an environment. Simplified models have been introduced for the analysis of the dynamics of robots. For example, a robot's whole body is abstracted into a point mass in the linear inverted pendulum (LIP) model [Kajita 1991], or into two point mass attributed to the head and the hip in the Spatial Double Inverted Pendulum Model [Xinjilefu 2009]. In addition to the LIP model, centroidal angular momentum has also been treated by the Angular Momentum Pendulum Model [Komura 2005], the Linear Inverted Pendulum Plus Flywheel Model [Pratt 2006], and the Reaction Mass Pendulum Model [Lee 2009]. In case of multiple non-coplanar contacts, the multiple-limbed robot with one mass point and multiple frictional point contacts has been used [Bretl 2008,Mansour 2011].

In addition to control approaches using simplified models, the control of whole-body behaviors to handle multiple objectives and constraints has received much attention. A variety of control systems have been proposed to handle complex tasks such as locomotion and object manipulation, subject to multiple constraints due to external perturbation forces, frictional contacts and geometric configurations. Many research works have been devoted to the development of multi-objective control systems. Some of them use optimization techniques [Abe 2007,Collette 2008,Ye 2010,Bouyarmane 2010] to find postures that optimize several objectives subject to multiple constraints. Some multi-task controllers treat tasks with different priority levels. Related work on prioritized control include the strategy of solving a sequence of quadratic programs [Kanoun 2009,Escande 2010,Kanoun 2011a] and the use of null-space projection methods [Si-

ciliano 1991,Sentis 2004,Khatib 2008,Sentis 2010,Raunhardt 2011]. One main contribution of this dissertation is a novel prioritized multi-objective control framework based on Jacobian-transpose method. We analyze the advantages and disadvantages of this control framework by a comparison with the prioritized control based on projection in Section 4.5.3.

1.4 Contributions

This dissertation studies virtual character control problem involving interactions with human operators, and motion control for performing multiple tasks in physics-based environments. Below are the contributions of this work:

Task-driven posture optimization (see Chapter 2 and [Liu 2012b]). We propose a posture optimization approach, which can be used in either the preparation for a manipulation task, or the evaluation of the feasibility of a task. This approach optimizes postures, including contact configurations, for the required task. The optimized postures ensure that the virtual character's end-effector can follow the desired manipulation trajectory. It also allows the character to apply manipulation forces as strongly as possible, and meanwhile to avoid foot slipping. Moreover, potential perturbation forces can be taken into account in the optimization to make postures more robust to perturbations. The realism of our approach is demonstrated with different types of manipulation tasks in Chapter 5.

Multi-objective control framework (see Chapter 3 and [Liu 2011b,Liu 2012a]). We design a multi-objective control framework, which adopts the Jacobian-transpose method and uses optimization techniques to compute appropriate motions for a virtual character. On one hand, this control framework allows an operator to interact with a virtual character in real-time; on the other hand, it allows the virtual character to compromise between following an operator's motions, and deciding by itself how to handle interactions with a physics-based virtual environment. This control framework allows a human operator to control a virtual character from a high level by just specifying an objective. So the operator is not responsible for providing appropriate postures. Task execution is realized by the multi-objective controller, which works at a low level.

Prioritized control with wrench bounds (see Chapter 4 and [Liu 2011a,

Liu 2012a]). We develop a new prioritized control approach, which is compatible with our multi-objective control framework based on Jacobian-transpose method. The novelty of this approach is that it can handle inequality constraints on a higher-priority task, and maintain the passivity of the system as well. An advantage of this method is that an operator does not need to compromise the balance of a virtual character when guiding its movements by real-time interactions.

The approaches developed in this dissertation are validated by experiments that are carried out on the simulator XDE [XDE] developed by CEA-LIST. XDE is a software environment that manages physics simulation in real time. Its physics simulation kernel handles multiple rigid and deformable bodies interacting with one another, and can achieve accurate collision detection.

Chapter 2

Task-driven Posture Optimization

Contents

This chapter presents a generic approach to find optimal postures, including contact positions, for manipulation tasks. It can be used in either the preparation for a task, or the evaluation of the feasibility of a task during planning stages. With such an approach, an animator can control a virtual character from a high level by just specifying a task, such as moving an object along a desired path to a desired position; the animator does not need to manually find optimal postures for the task. For each task, an optimization problem is solved, which considers not only geometric and kinematic constraints, but also force and moment constraints. The optimized postures allow the virtual character to apply manipulation forces as strongly as possible, and meanwhile to avoid foot slipping. Moreover, potential perturbation forces can be taken into account in the optimization to make postures more robust. In this chapter, results using this approach are demonstrated on a simplified model of a character, by tasks of manipulating objects along different paths and towards different directions. The realism of this approach is further demonstrated on

a complete virtual character in Chapter 5, where posture optimization results are used by our multi-objective controller to improve task performance.

2.1 Introduction

A prime functionality of a virtual character is to perform manipulation tasks. The choice of postures can impact the possibility of fulfilling a task successfully. Here we call *posture* a set of body configurations including contact positions. Optimal postures usually vary from task to task. For example, the virtual character may have to lean forward to push an object, but lean backward to pull it; the feet may have to be separated from each other to be able to generate manipulation forces that are sufficiently strong. Even for the same kind of tasks, for example pushing, the optimal posture should be adjusted to adapt to different object's physical properties. Moreover, available contact positions are sometimes restricted due to environment constraints. For example, the virtual character may need to choose foot positions that allow the end-effector to manipulate an object without moving the feet. These questions suggest that before performing a task, it is important and beneficial for the virtual character to choose postures that are optimal for the task.

In the context of computer animation where motion capture has become an essential technique, an operator's postures can be taken as references for the virtual character. Captured motions are lifelike, but they need to be adjusted to handle manipulation forces and to deal with disturbances. This is because the operator usually does not really manipulate objects and thus cannot sense interaction forces between the virtual character and the virtual environment. Moreover, the operator may not be skillful enough for providing suitable reference postures for some tasks. Consequently, his postures can be inappropriate for the virtual character to balance the interaction forces or to improve task performances. Many existing methods focus on the generation of foot contact positions for locomotion tasks. The choice of contact positions with the purpose of improving manipulation task performances by taking into account contact forces remains a challenge.

This chapter introduces a generic approach that can automatically find optimal postures for a wide variety of manipulation tasks. For each manipulation task, a constrained

optimization problem is solved off-line to find a sequence of optimal postures associated with a desired manipulation path, in the neighborhood of a given initial posture. The optimization problem is based on a simplified model of the character. This simplified model takes into account interaction forces with the environment and the kinematic relations between control frames. Here *control frames* are some coordinate frames attached to the character's body, the positions of which are to be optimized. Once a solution is found, one can use a motion controller to make the character adjust the contact positions, then to perform object manipulation by following the desired motions of the control frames and by applying desired forces. Our approach considers quasi-static cases where dynamic effects can be ignored; therefore, we suggest applying a quasi-static task controller for object manipulation, such as the one described in [Liu 2011b].

The main contributions of this approach are as follows:

- It is a generic posture optimization approach that couples **geometric and kinematic constraints (G-K constraints)** with **force and moment constraints (F-M constraints)**.
- It can improve task performance by choosing suitable postures in a preparation stage before actually performing the task. Contact configurations for manipulation tasks are optimized for user-specified manipulation paths and forces.
- It deals with the redundancy of poses, and can make contact positions as robust as possible. The structure of our posture optimization problem allows us to take precaution against mechanical interactions and possible perturbations. By adopting the optimized postures, the risk of failures either due to poor postures or due to perturbations can be greatly decreased.

2.2 Related work

Task-based constraints should take into account kinematic constraints related to the character's body structure and geometric relations between the character and the environment. For example, the character may need to find postures that allow its hand to move an object along a desired trajectory without violating the constraint of its body structure. The constraints related to a body structure can be formulated based on forward kinematics,

which provides a mapping between a body frame and the joint angles. One example is the virtual kinematic loop equation [Smits 2010], which can be used to force the frame of a link to coincide with another frame in the environment. Researchers in robotics have adopted such a constraint in the path tracking of robot manipulators and wheeled robots [de Schutter 2005,De Schutter 2007,Stilman 2010]. Our approach applies this constraint on a virtual character to find kinematically feasible motions for manipulation tasks. To perform specified interactions with the environment, some kinematic-based motion editing approaches modify input human motions [Peinado 2009,Jain 2009a]. In [Liu 2002], environmental restrictions are represented as positional and sliding constraints, and linear and angular momentum constraints are used to improve the realism of motions. Constrained inverse kinematics has been combined with a database of example postures to synthesize motions that satisfy a set of G-K constraints for manipulation tasks [Yamane 2004]. However, in a physics-based simulation environment, these approaches are limited when the character needs to react to interaction forces during manipulation tasks; and moreover, approaches based on a motion database usually cannot generate certain behaviors to handle interactions if such behaviors are not included in the motion database.

To adapt output motions to interaction forces, F-M constraints should be taken into account. For example, the equilibrium of forces and moments should be considered to ensure balance; and contact forces should be handled for manipulation tasks, because the character's body is under-actuated and it needs to use contact forces to perform desired motions. In some optimization based motion synthesis approaches, physical constraints based on forces and torques are included in the optimization to ensure physical realism [Fang 2003,Jain 2009b]. F-M constraints have been considered in many task control frameworks [Khatib 2004,Abe 2007,Collette 2007,Liu 2011b], where the desired foot contact positions are either given a priori, or computed without considering interaction forces during object manipulation. Our posture optimization can be considered as a preparation step that can be executed before applying these task controllers. In fact, our approach provides these task controllers with a pre-computed solution of suitable postures for given manipulation tasks. These postures, including contact positions such as foot positions, are optimized with respect to user-specified manipulation paths and forces. We achieve this by taking into account F-M constraints to handle physical interactions, and G-K constraints

to generate kinematically feasible motions with environment awareness.

The configuration of contacts is a major aspect that is focused on in this chapter. A support polygon reshaping approach has been proposed in [Yoshida 2006], the idea of which is to first try to reach the target with an initial support polygon, and then reshape the support polygon according to the feedback task error. An approach to plan foot placements according to kinematic tasks has been described in [Kanoun 2011b]. Compared with these approaches, ours is more general in that besides G-K constraints, we also take into account F-M constraints in the optimization of contact positions. Moreover, under many circumstances, the feet of the character are fixed when its hands are manipulating objects along a segment of manipulation path [Saab 2009,Hauser 2011]. Our approach can provide a support polygon that is suitable for a segment of end-effector motion path instead of for only one fixed end-effector task target; therefore the character will not need to adjust its foot positions frequently during task execution.

Besides, we also optimize contact positions to generate robust postures with respect to perturbation forces. An optimal control which allows the adaptation of walking motions to physical perturbations has been proposed in [Ye 2010]. Contact forces are first generated off-line to reproduce reference motions, and then adjusted on-line to maintain contacts and balance during perturbations. But such contact forces that satisfy current contact configuration may not exist. As our approach considers manipulation tasks with the desired interaction forces known a priori, contact forces and contact configurations are optimized simultaneously before task execution. Possible perturbation forces can also be taken into account in our optimization to make the solution more robust. Moreover, kinematic relations between contacts and control frames are not considered in [Ye 2010], which may generate kinematically unfeasible motions. Such kind of relations is taken into account in our approach.

2.3 Posture optimization

The goal of this section is to formulate an optimization problem, which will be solved to find optimal postures in the neighborhood of a given one. Such postures should allow the hand to follow a manipulation path defined by the task, and to apply sufficiently

strong interaction forces on the object. We seek to improve the robustness of the posture against external perturbation forces, and in addition, to make the postures as comfortable as possible. Besides, the optimization result also provides possible contact forces on the feet and the maximum allowable interaction force between the end-effector and the object.

2.3.1 Simplified model

Our posture optimization problem uses a simplified model, which considers essential elements of the character, such as contact forces, the positions of the CoM and some other control frames, as well as the kinematic relations between these control frames. This simplified model consists of a punctual mass (m) at the CoM, one massless back, one massless arm and two massless legs (see Fig.2.1). The model has 15 degrees of freedom (DoF), including 6 root DoF (3 translational DoF and 3 rotational DoF), 2 DoF for each hip and shoulder and 1 DoF for each knee and elbow. It has two contact points, one on

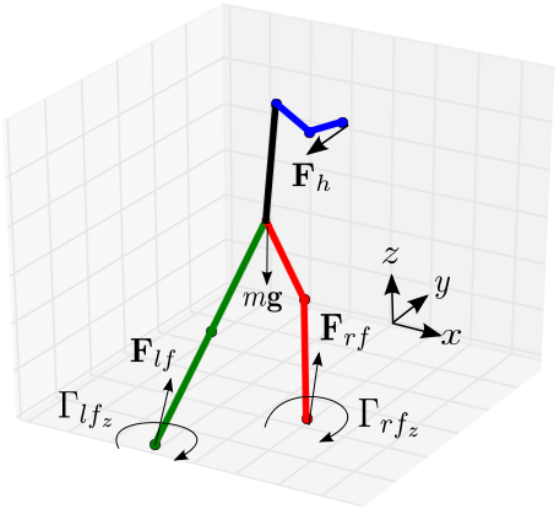


Figure 2.1: Simplified model.

each foot; and it has one end-effector, which is the hand. The contacts on each foot and hand are abstracted into one frictional point contact as in [Bretl 2008], which means only the net contact force between each body segment and the environment is considered. These simplifications help to reduce the dimension of the optimization problem while retaining the important characteristics of the interaction model.

The following notation is used in this chapter.

- All the position vectors $\mathbf{X}$ are defined with respect to a global reference frame with axis x, y, and z. The z axis points upwards.
- The l-th joint angle is denoted as $\mathbf{q}_l$.
- The gravity force applied at the CoM is denoted as $\mathbf{F}_G$.
- Each frame j on the body of the virtual character is generally denoted by subscript j. More specifically, frames are denoted by subscripts: *com* for the CoM, *lf* and *rf* for the left and right foot respectively, h for the hand, and g for the ground.
- The upper and lower limits of a variable a are denoted by a^U and a^L respectively.

The manipulation path is discretized into sampled points. A term associated with a discretized point i is denoted by the superscript i. The desired position of the object at each discretized point i is denoted as $\mathbf{X}^i_{\text{obj}}$. The hand force at the position $\mathbf{X}^i_{\text{obj}}$ is defined as $\mathbf{F}^i_h = k^i \hat{\mathbf{F}}^i_h$, where $\hat{\mathbf{F}}^i_h$ is a unit vector indicating the desired force direction at point i, and k^i is its magnitude.

2.3.2 Optimization with respect to one discretized point on the manipulation path

For clarity, we first describe a posture optimization problem with respect to one pair of object position and desired manipulation force direction $(\mathbf{X}^i_{\text{obj}}, \hat{\mathbf{F}}^i_h)$. The optimization takes into account the objectives of increasing the maximum allowable manipulation forces, minimizing the risk of foot slipping, and reducing joint discomfort, subject to all the G-K and F-M constraints.

The set of the optimization variables is defined as

$$\Theta^i = \left\{\mathbf{X}^i_{\text{com}}, \mathbf{q}^i, \mathbf{F}^i_{\text{lf}}, \mathbf{F}^i_{\text{rf}}, \mathbf{\Gamma}^i_{\text{lf}_z}, \mathbf{\Gamma}^i_{\text{rf}_z}, k^i\right\} \tag{2.1}$$

The optimization problem is written as follows:

$$\begin{aligned} &\min_{\Theta^i}(w_h G^i_h(\Theta^i) + w_f G^i_f(\Theta^i) + w_q G^i_q(\Theta^i)) \\ &\text{s.t. } \Psi^i_{\text{GK}}(\mathbf{X}^i_{\text{com}}, \mathbf{q}^i) \\ &\quad \Psi^i_{\text{FM}}(\Theta^i) \end{aligned} \tag{2.2}$$

where G_h, G_f and G_q are the objectives, and Ψ_{GK} and Ψ_{FM} are G-K and F-M constraints respectively. The optimization weights w are chosen based on different task requirements.

We will discuss how to make the choice of these weights in section 5.3.

2.3.2.1 Objectives

To improve the manipulation ability, the hand force magnitude along the given direction is maximized by setting the following hand force objective.

$$G_h^i(k^i) = -k^i \tag{2.3}$$

To avoid foot slipping, the foot contact forces should remain inside their friction cones. This non-sliding constraint will be referred to later in this chapter. As this constraint is not sufficient to fully determine the tangential foot contact forces, these forces are minimized by the following objective function, which helps to reduce the risk of foot slipping.

$$G_f^i(\mathbf{F}_{\text{lf}}^i, \mathbf{F}_{\text{rf}}^i) = \frac{1}{2}\left\|\mathbf{S}_{\text{lf}}\mathbf{F}_{\text{lf}}^i\right\|^2 + \frac{1}{2}\left\|\mathbf{S}_{\text{rf}}\mathbf{F}_{\text{rf}}^i\right\|^2 \tag{2.4}$$

where $\mathbf{S}_{\text{lf}}$ and $\mathbf{S}_{\text{rf}}$ denote matrices to select the directions of the tangential friction forces.

A real human always intend to reduce joint discomfort during manipulation tasks. An objective function of joint discomfort G_q is used so as to imitate such human behaviors. The discomfort measure [Yang 2004,Ma 2009] is applied here. The objective function is defined as follows.

$$\begin{aligned}
G_q^i(q^i) &= \sum_{l=1}^{DoF}\left[\phi_l(\Delta q_l^n)^2 + QU_l^i + QL_l^i\right] \\
\Delta q_l^n &= \frac{q_l^i - q_l^N}{q_l^U - q_l^L} \\
QU_l^i &= (0.5cos\frac{\frac{3\pi}{2}(q_l^U - q_l^i)}{q_l^U - q_l^L} + 0.5)^{100} \\
QL_l^i &= (0.5cos\frac{\frac{3\pi}{2}(q_l^i - q_l^L)}{q_l^U - q_l^L} + 0.5)^{100}
\end{aligned} \tag{2.5}$$

This objective guides the optimization to choose joint angles based on their neutral values and limits. It attempts to push joint angles q^i away from their upper limits q^U and lower limits q^L, and pull them towards a neutral value q^N, so as to increase posture comfort level. As mentioned in [Yang 2004], the concept behind the discomfort measure is to enhance the preference of using certain joints to fulfill a motion task, by regulating the joint weight ϕ_l.

We will show in section 5.3 that a careful choice of the value of ϕ_l helps to improve the behaviors of the virtual character, making them closer to those of a real human.

2.3.2.2 Geometric and kinematic constraints

Our G-K constraints take into account the geometric relations between the character and the virtual environment, as well as the kinematic relations between the control frames. An example of the constraints Ψ^i_{GK} is listed below.

Joint angles should respect joint limit constraints.

$$\mathbf{q}^L \leq \mathbf{q}^i \leq \mathbf{q}^U \tag{2.6}$$

We search for position solutions within a constrained region of interest, which is a polygon around the object.

$$\begin{aligned} &\mathbf{A}_{\mathrm{com}}\mathbf{X}^i_{\mathrm{com}} + \mathbf{b}_{\mathrm{com}} \leq \mathbf{d}_{\mathrm{com}} \\ &\mathbf{A}_{\mathrm{lf}}\mathbf{X}_{\mathrm{lf}x,y}(\mathbf{X}^i_{\mathrm{com}}, \mathbf{q}^i) + \mathbf{b}_{\mathrm{lf}} \leq \mathbf{d}_{\mathrm{lf}} \\ &\mathbf{A}_{\mathrm{rf}}\mathbf{X}_{\mathrm{rf}x,y}(\mathbf{X}^i_{\mathrm{com}}, \mathbf{q}^i) + \mathbf{b}_{\mathrm{rf}} \leq \mathbf{d}_{\mathrm{rf}} \end{aligned} \tag{2.7}$$

The hand position is constrained to point $\mathbf{X}^i_{\mathrm{obj}}$, which implies that the hand moves along the desired manipulation path, as the object does.

$$\mathbf{X}_h(\mathbf{X}^i_{\mathrm{com}}, \mathbf{q}^i) - \mathbf{X}^i_{\mathrm{obj}} = \mathbf{0} \tag{2.8}$$

The following constraint is imposed to prevent the feet from overlapping each other. The distance between the feet is kept lager than a minimum value.

$$\psi(\mathbf{X}_{\mathrm{lf}x,y}(\mathbf{X}^i_{\mathrm{com}}, \mathbf{q}^i), \mathbf{X}_{\mathrm{rf}x,y}(\mathbf{X}^i_{\mathrm{com}}, \mathbf{q}^i)) \geq d_f \tag{2.9}$$

where $\psi(\mathbf{X}_1, \mathbf{X}_2)$ denotes the distance between $\mathbf{X}_1$ and $\mathbf{X}_2$. The angle between the facing direction of the character and the direction of the object is constrained in (2.10), where ϑ denotes the angle between the two vectors.

$$\vartheta(\mathbf{X}^i_{\mathrm{com}}, \mathbf{q}^i, \mathbf{X}^i_{\mathrm{obj}}) \leq d_\vartheta \tag{2.10}$$

The positions of some control frames j may have to respect some additional geometric constraints (2.11) in constrained environments. For example, the optimal positions of the feet should not penetrate into an object.

$$\psi(\mathbf{X}_{\mathrm{obj}}, \mathbf{X}_j(\mathbf{X}^i_{\mathrm{com}}, \mathbf{q}^i)) \geq 0 \tag{2.11}$$

The contacts between the feet and the ground should be maintained (2.12):

$$\mathbf{X}_{\mathrm{lf,rf}_z}(\mathbf{X}^i_{\mathrm{com}}, \mathbf{q}^i) - \mathbf{X}_{g_z} = 0 \tag{2.12}$$

Since the position of each control frame can be obtained by forward kinematics, they are expressed as a function of the CoM position $\mathbf{X}^i_{\mathrm{com}}$ and joint angles $\mathbf{q}^i$. As a result, constraints due to the skeleton structure are implicitly included in these G-K constraints.

2.3.2.3 Force and moment constraints

This chapter is interested in manipulation tasks where the major perturbation comes from mechanical interactions. The character should be able to keep its balance under external contact forces. To achieve this goal, the following F-M constraints Ψ^i_{FM} are imposed. Only the quasi-static cases are considered here, and the dynamic effects such as acceleration are neglected.

To maintain the static equilibrium, the constraint of force and moment balance (2.13) is imposed.

$$\begin{aligned} &\mathbf{F}^i_{\mathrm{lf}} + \mathbf{F}^i_{\mathrm{rf}} + \mathbf{F}^i_h(k^i) + \mathbf{F}_G = \mathbf{0} \\ &\mathbf{X}_{\mathrm{lf}}(\mathbf{X}^i_{\mathrm{com}}, \mathbf{q}^i) \times \mathbf{F}^i_{\mathrm{lf}} + \mathbf{X}_{\mathrm{rf}}(\mathbf{X}^i_{\mathrm{com}}, \mathbf{q}^i) \times \mathbf{F}^i_{\mathrm{rf}} \\ &\quad + \mathbf{X}_h(\mathbf{X}^i_{\mathrm{com}}, \mathbf{q}^i) \times \mathbf{F}^i_h(k^i) + \mathbf{X}^i_{\mathrm{com}} \times \mathbf{F}_G + \mathbf{\Gamma}^i_{\mathrm{lf}_z} + \mathbf{\Gamma}^i_{\mathrm{rf}_z} = \mathbf{0} \end{aligned} \tag{2.13}$$

In order to avoid foot slipping, each foot contact force is constrained to remain inside a friction cone in (2.14).

$$\left\| \left[\mathbf{F}^i_{\mathrm{lf,rf}_x}, \mathbf{F}^i_{\mathrm{lf,rf}_y} \right]^T \right\| \leq \mu \left\| \mathbf{F}^i_{\mathrm{lf,rf}_z} \right\| \tag{2.14}$$

with μ denoting the friction coefficient between the feet and the ground. The hand force magnitude is constrained as follows.

$$k^L \leq k^i \leq k^U \tag{2.15}$$

The lower bound k^L is defined by the task. It stands for the minimum magnitude of the interaction force that is necessary for performing the object manipulation.

2.3.3 Optimization with respect to a manipulation path

Given a manipulation path and the desired force directions along the path, the whole posture optimization problem is solved with respect to each discretized point i. We want

the foot positions to be fixed during the manipulation task (Figure 2.2), so the following constraints are used between each discretized step.

$$\mathbf{X}_{\mathrm{lf,rf}}(\mathbf{X}^i_{\mathrm{com}}, \mathbf{q}^i) - \mathbf{X}_{\mathrm{lf,rf}}(\mathbf{X}^{i-1}_{\mathrm{com}}, \mathbf{q}^{i-1}) = \mathbf{0}, i > 2. \tag{2.16}$$

In this way, the final solution of foot positions satisfy not only the constraints associated with a local point i, but also those for the whole motion path. It should be noticed that we did not impose similar constraints on the CoM position and joint angles. Therefore the virtual character is allowed to move its body and change its posture during manipulation, even though its feet are fixed.

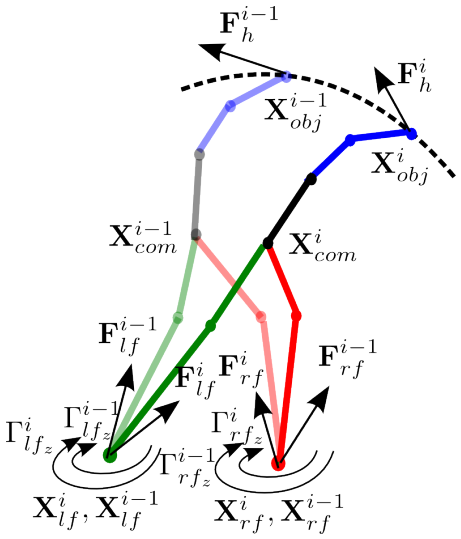


Figure 2.2: Postures along a motion path with fixed foot positions.

Moreover, the minimum value of k^i is maximized to maximize the hand force along the whole path. Hence, instead of using (2.3) as the hand force objective function, the following one is used for the whole path:

$$G_h(\{k^i\}) = -\min_i \{k^i\} \tag{2.17}$$

An advantage of our posture optimization is that it can be used to improve the robustness by taking account of perturbations in the optimization problem. This can be done by adding more pairs of hand position and hand force direction. For example, at point i of the original motion path, some perturbations ($\delta\mathbf{X}_{\mathrm{obj}}$, $\delta\mathbf{F}_h$) of different magnitudes and

directions can be added to $\mathbf{X}^i_{\text{obj}}$ and $\mathbf{F}^i_h$.

$$\begin{aligned} \tilde{\mathbf{X}}^i_{\text{obj}} &= \mathbf{X}^i_{\text{obj}} + \delta\mathbf{X}_{\text{obj}} \\ \tilde{\mathbf{F}}^i_h &= \mathbf{F}^i_h + \delta\mathbf{F}_h \end{aligned} \tag{2.18}$$

The optimization problem is solved with respect to a set of possible hand positions $\tilde{\mathbf{X}}^i_{\text{obj}}$ and force directions $\tilde{\mathbf{F}}^i_h$, so that the posture solution can better cope with the perturbations.

The optimization problem is summarized as follows:

$$\begin{aligned} \min_{\Theta=\bigcup_i \Theta^i} \; & w_h G_h + \sum_i (w_f G^i_f + w_q G^i_q) \\ \text{s.t.} \; & \left\{\Psi^i_{\text{GK}}(\mathbf{X}^i_{\text{com}}, \mathbf{q}^i)\right\} \\ & \left\{\Psi^i_{\text{FM}}(\Theta^i)\right\} \end{aligned} \tag{2.19}$$

The above optimization problem contains several equality and inequality constraints, most of which are nonlinear. We solve it by using CFSQP algorithm [C. Lawrence 1997].

Note that the optimization problem presented here is just an example to explain the idea. It can be generalized to handle other problems with different contact configurations, such as a character moving one foot using both hands and the other foot as fixed contacts.

2.4 Results

The approach presented in the above section has been tested on the simplified model. The following three types of experiments (Exp_1, Exp_2 and Exp_3) are considered. A solution of postures and especially foot positions are to be found.

– Exp_1:

Find a solution which allows the robot to move its hand between two points, $\mathbf{X}^1_{\text{obj}}$ and $\mathbf{X}^2_{\text{obj}}$, and to maximize the manipulation force along *one given direction* $\hat{\mathbf{F}}_h$. This problem can be written as $\Pi(\{\Theta^1, \Theta^2\} \mid \left\{(\mathbf{X}^1_{\text{obj}}, \hat{\mathbf{F}}_h), (\mathbf{X}^2_{\text{obj}}, \hat{\mathbf{F}}_h)\right\})$.

– Exp_2:

Find a solution which allows the robot to move its hand between two points, $\mathbf{X}^1_{\text{obj}}$ and $\mathbf{X}^2_{\text{obj}}$, and to maximize the manipulation force in *two inverted directions*, $\hat{\mathbf{F}}_h$ and $-\hat{\mathbf{F}}_h$. This problem can be written as $\Pi(\{\Theta^1, \Theta^2\} \mid \left\{(\mathbf{X}^1_{\text{obj}}, \hat{\mathbf{F}}_h), (\mathbf{X}^2_{\text{obj}}, -\hat{\mathbf{F}}_h)\right\})$.

– Exp_3:

Find solution which allows the robot to move its hand along *a given curve* with n discretized points $\{\mathbf{X}^i_{\text{obj}}, \forall i \in N = \{1, 2, ..., n\}\}$, and to maximize the manipulation force, which points to *several different directions* throughout the path. This problem can be written as $\Pi(\{\Theta^i, \forall i \in N\} \mid \{(\mathbf{X}^i_{\text{obj}}, \hat{\mathbf{F}}^i_h), \forall i \in N\})$.

We conducted several experiments for these types of problems. The results of postures are shown in Fig.2.3 - Fig.2.5 for problem type Exp_1, in Fig.2.6 - Fig.2.8 for problem type Exp_2, and in Fig.2.9 for problem type Exp_3.

The results of optimization demonstrates that the above-mentioned optimization problem can be successfully resolved based on G-K constraints and F-M constraints. The computation time depends on the problem to solve, especially the number of discretized points. According to the data obtained from Exp_1, where there are 2 discretized points, it takes a duration of time ranging from 0.08s to 2.06s, with an average of 1.33s. Generally it takes less than one minute if the number of points is smaller than 5.

(a) pull rightward

(b) pull leftward

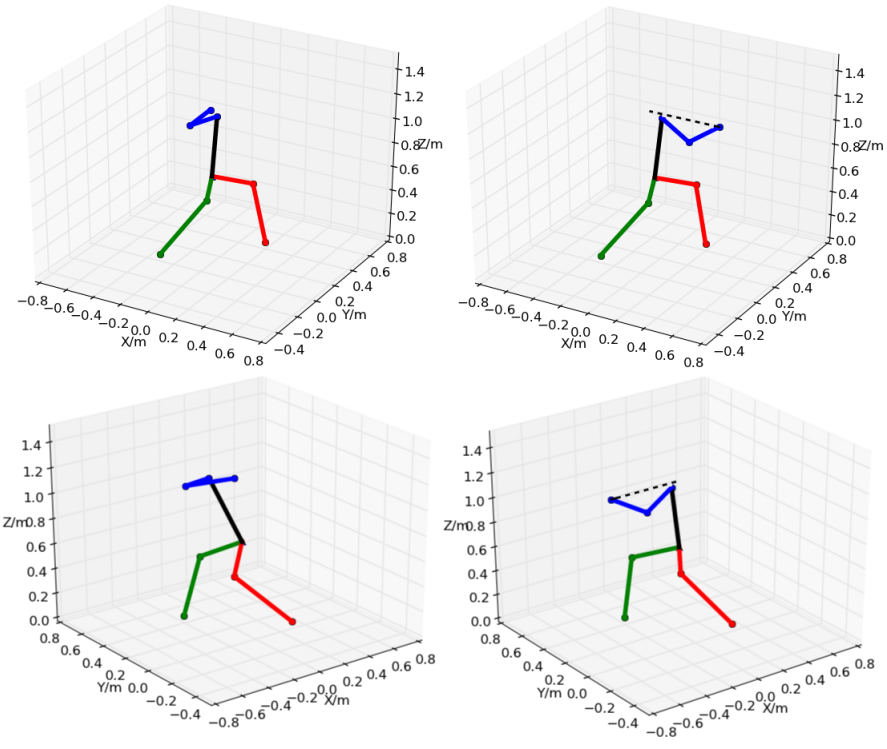


Figure 2.3: Results of postures for Exp_1. Each row corresponds to one manipulation direction. Figures in the left and right column correspond to the start states and the end states respectively. The path is shown in a black dotted line.

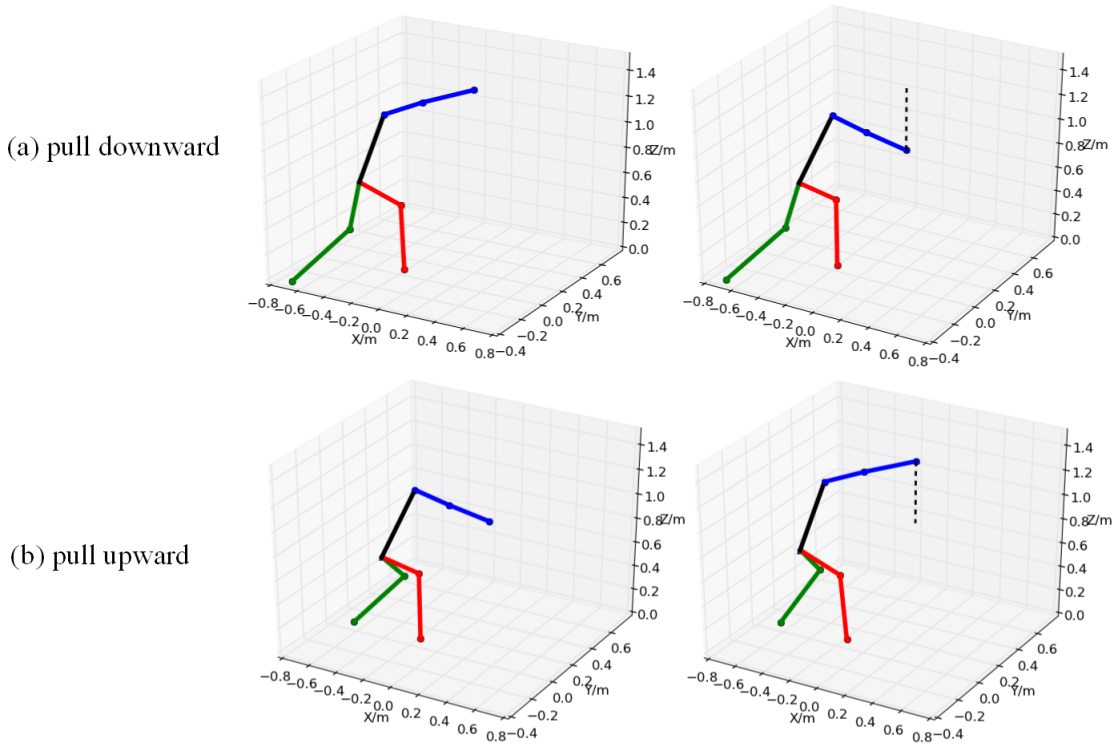


Figure 2.4: Results of postures for Exp_1. Each row corresponds to one manipulation direction. Figures in the left and right column correspond to the start states and the end states respectively. The path is shown in a black dotted line.

The approach is capable of providing suitable postures with respect to a variety of manipulation tasks. The obtained postures are different from task to task, changing not only in favor of different motion paths, but also of different interaction force directions. In Exp_1, the postures are adapted to the desired interaction force direction. For example when pushing an object (Fig.2.5(b)), the body leans forward to be able to push more strongly. The robot deliberately increases the feet distance along the same direction of the given motion path or manipulation force direction. This makes sense since separating the feet along a direction helps to generate a more robust posture against the interaction force along that direction. We can observe similar postures when a real human performs the same tasks.

The difference between Exp_1 and Exp_2 is that the former is an optimization based on one desired force direction, whereas the latter is based on two inverted force directions.

(a) pull backward

(b) push forward

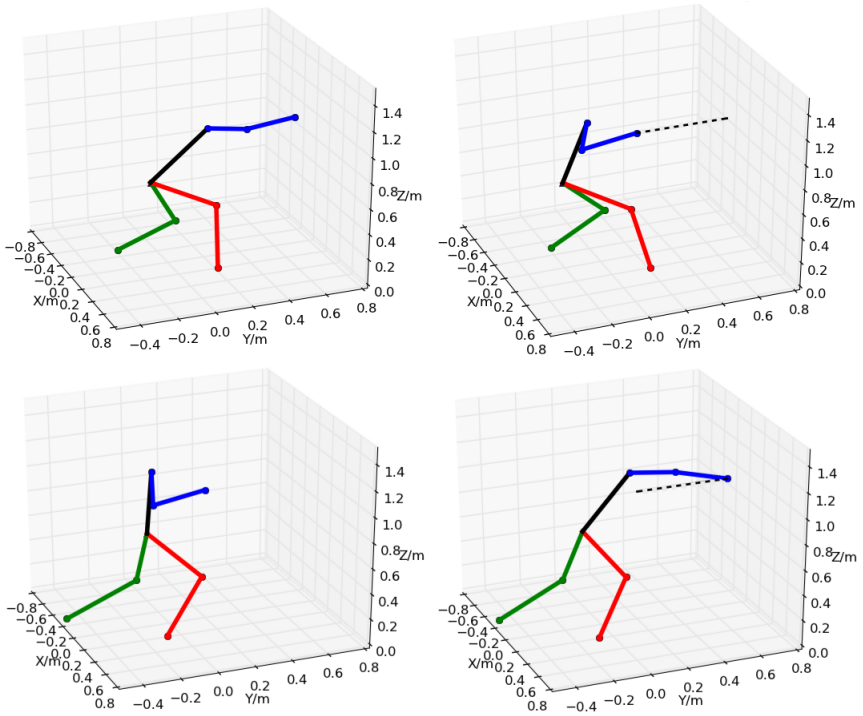


Figure 2.5: Results of postures for Exp_1. Each row corresponds to one manipulation direction. Figures in the left and right column correspond to the start states and the end states respectively. The path is shown in a black dotted line.

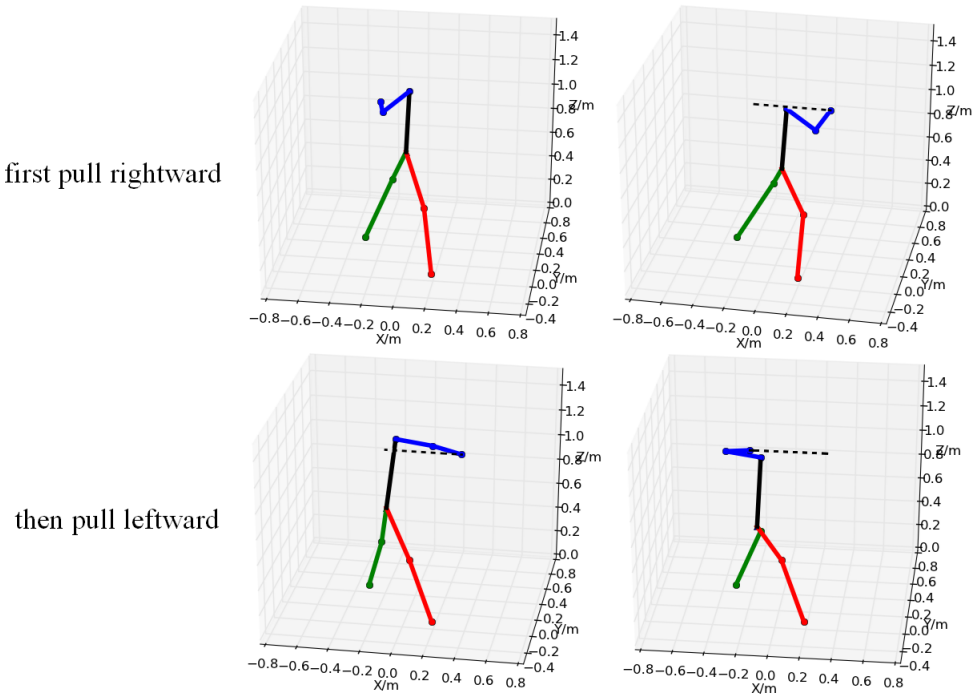


Figure 2.6: Results of postures for Exp_2. The robot is required to first pull an object to the right then pull it back to the left. The path is shown in a black dotted line. The hand positions at both ends are the same as in Fig.2.3.

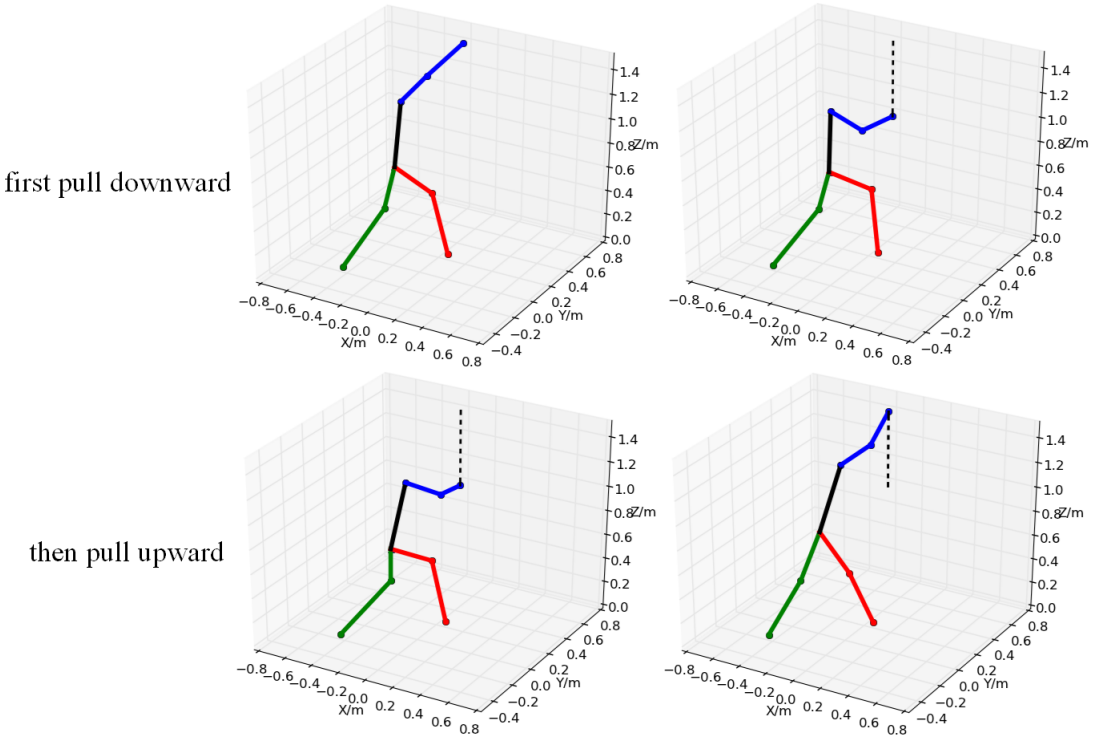


Figure 2.7: Results of postures for Exp_2. The robot is required to first pull an object down then pull it up. The path is shown in a black dotted line. The hand positions at both ends are the same as in Fig.2.4.

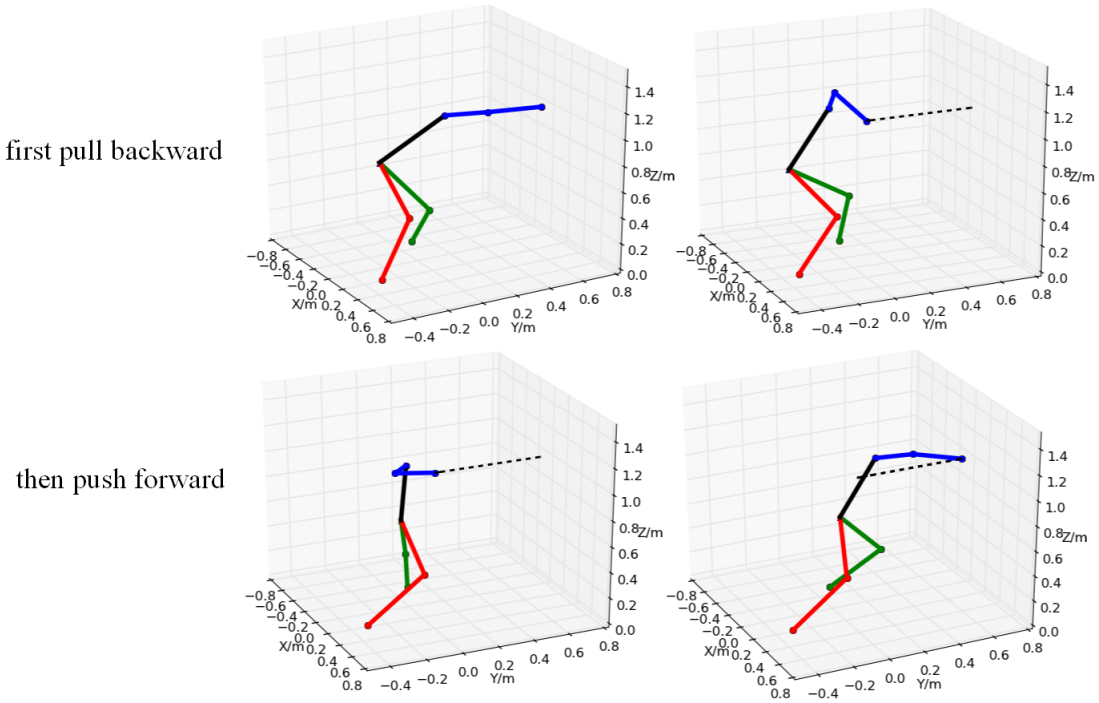


Figure 2.8: Results of postures for Exp_2. The robot is required to first pull an object back then push it forward. The path is shown in a black dotted line. The hand positions at both ends are the same as in Fig.2.5.

Note that the motion path in Fig.2.6 is the same as in Fig.2.3, but the foot positions are different. In Fig.2.3 (a) and (b), each foot contact configuration is optimal for one manipulation force direction. However, the foot positions in Fig.2.6 are more adaptable to manipulation forces in both directions. Similar results can be found in Fig.2.7 and Fig.2.8.

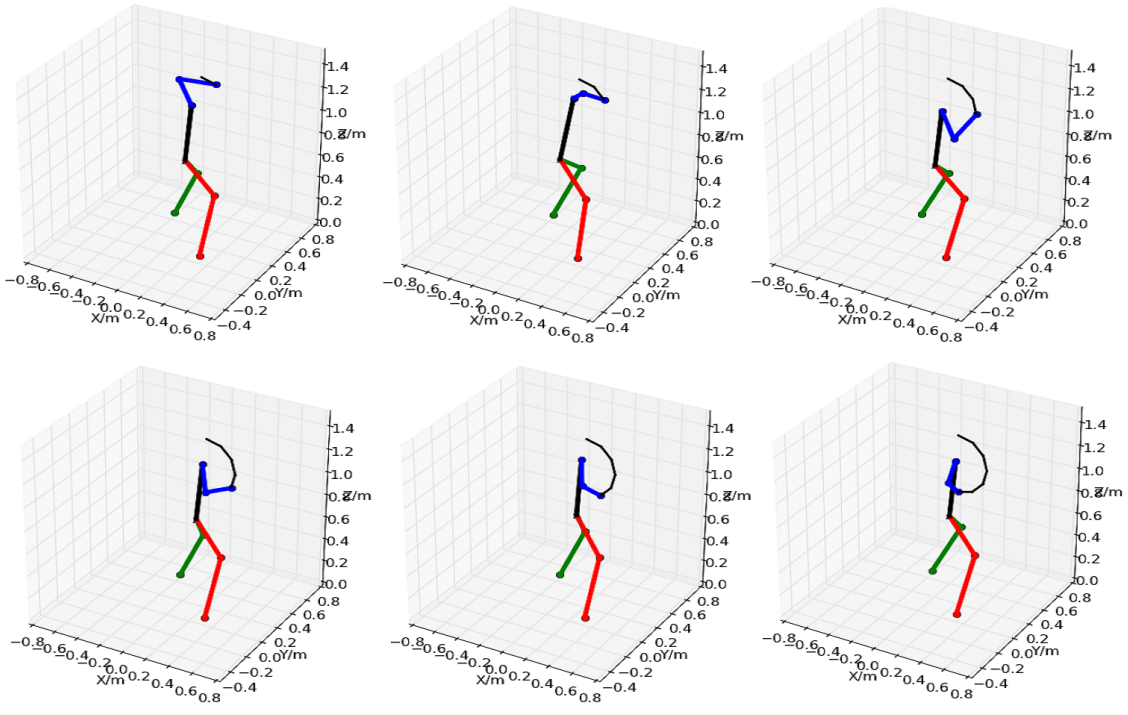


Figure 2.9: Results of postures for Exp_3. The given curved path is shown in black lines. The desired force direction is tangent to the curve at each point.

Results of Exp_3 suggest that the proposed approach is capable of finding suitable postures enabling the hand to track a motion path which is more complex than a straight line, and to apply forces towards different directions along the path as well (see Fig.2.9).

The maximum allowed manipulation force along a motion path provided in the optimization results may help us to better evaluate a task before executing it. On one hand, if the task is unrealistic and will conflict with some constraints, then the optimization will provide no solution; on the other hand, the optimization solution can provide us with an idea of the maximum manipulation force that the character can apply on an object along a motion path. The upper limit of the force magnitude used in each of our experiments is

$150N$. We found that in most cases (90 percent) the obtained maximum allowable force magnitude can attain this upper limit.

As is mentioned in Section 2.3.3, an advantage of this planning method is that we can use it to make the standing position and the shape of support polygon more robust to perturbations. The result of such implementation will be shown in Section 5.3.5 to demonstrate how the optimized postures can better cope with perturbations during manipulation.

The objective weights are chosen according to task requirements. If the character has to apply a strong manipulation force, for example, moving a heavy object, then the weight w_h should be set to a high value to enhance the hand force objective, and to ensure that the maximum allowable manipulation forces are sufficient for the task. If we want to reinforce non-sliding contacts on the feet, then w_f should be assigned with a higher value to reduce tangential contact forces on the feet; however, the maximum allowable value of k might be limited as a compromise. The objective weights used in our experiments are: $w_h = 1000$, $w_f = 1$, and $w_q = 500$.

2.5 Limitations

The current approach has a few limitations.

First, since the posture optimization problem that we are dealing with is not convex, several local minima may exist. The solution of our posture optimization is a local optimum in the neighborhood of an initial value; and the global optimum might be drastically different. This is because the CFSQP algorithm that we use to solve our problem is based on derivatives, which leads to a local minimum. However, we choose to use a derivative-based optimization algorithm because it converges the fastest. One possible solution to improve the posture solution is to build a database of captured motion for different kinds of manipulation tasks, so as to provide natural and lifelike initial postures.

Second, the computation time for solving the optimization problem is sensible to the given task, especially the complexity of the motion path. Currently we first apply posture optimization off-line. Then use the optimization results in the on-line task controller.

Moreover, the current posture optimization might not always be able to find an optimal

solution, especially when the manipulation path is too long for the optimization to find a suitable contact configuration that supports the whole path. We plan to handle this problem by developing some automatic segmentation techniques; so that the path can be divided into several segments automatically, and posture optimization will be executed segment by segment.

2.6 Conclusion

We have developed a generic approach to find optimal postures, especially foot positions, for object manipulation tasks. The optimized posture can enable the end-effector to follow given manipulation path while applying the maximum manipulation forces without causing foot slipping and balance problems. Besides, constraints such as joint limits, non-sliding contacts, and geometrical relations with the environment can be satisfied. The results of our experiments suggest that the proposed posture optimization problem based on both G-K constraints and F-M constraints can be numerically solved for a wide variety of tasks. The obtained postures are different from task to task, changing not only in favor of different motion paths, but also for different interaction forces.

Future work will study how to reduce the computation time when there are a large number of discretized points along a complex motion path. We also plan to improve our optimization framework by taking into account objectives concerning joint torques as in [Boulic 1996,Harada 2006]. Moreover, we can make our approach more generic by optimizing the trajectory of the center of pressure (CoP) as well. So for each contact, the CoP will be optimized and will be allowed to move inside the support polygon instead of being fixed. To realize this, the foot size and the admissible shape of the support polygon should also be taken into account in the optimization.

Chapter 3

A Multi-objective Control Framework

Contents

This chapter presents a new control framework for virtual characters in a physics-based virtual environment. This framework combines multi-objective control with motion capture techniques. Each motion tracking task is associated with a task wrench. An optimization problem is solved to compute optimal task wrenches. Finally, joint torques are computed using the optimal task wrenches. This control framework allows the virtual character to generate appropriate motions to handle interactions with the virtual environment, rather than to simply emulate captured motions. The effectiveness of our approach is demonstrated by a virtual character performing manipulation tasks.

3.1 Introduction

Virtual characters usually need to interact with human operators and with virtual environments. They should react to operator's instructions and simulated events. Motion capture has become an essential technique in the control of virtual characters. It is traditionally used to guide the motions of a virtual character by virtual springs [Lee 2010]. Many motion correction techniques that are based on prerecorded motions [Callennec 2006,Shapiro 2008,Jain 2009a] have shown good results, whereas our work studies real-time interactions where an operator can interact in an unpredictable way.

There are two main difficulties in this research topic. First, since a virtual character's body is under-actuated, it should make contacts with the environment and use contact forces to perform desired movements. However, the operator's postures can be inappropriate for the virtual character to deal with contact forces during task execution. This is because, in order to simplify the interaction system between the operator and the virtual character and to make the operator more comfortable, it is preferable to use a minimum of equipments for the interaction. To achieve a low-immersion virtual reality environment, we choose to use only a motion capture system for the interaction, without using force sensors. As a result, the virtual character should be endowed with certain autonomy to improve its performance automatically, in order to compensate for the fact that the operator is not completely immersed into the virtual environment and cannot adjust his postures according to the feedback of the forces applied on the virtual character. Consider the scenario where a virtual character pushes an obstacle (Fig.3.1) for example. The operator just sends out the intention of pushing by reaching out his hands, but the virtual character may have to automatically lean towards the obstacle to push it while the operator cannot, because the virtual obstacle does not exist in the world of the operator. Hence captured motions should be adjusted to be more suitable to handle external contact forces during interactions with the environment. Second, virtual characters are complex systems, with a high number of degrees of freedom. They can have many postures which help to accomplish the same task. The controller must find the best solution among all the possible postures.

The contributions of this chapter is the development of a multi-objective control frame-

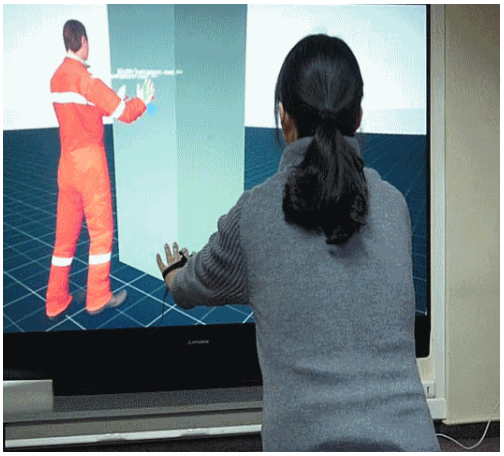


Figure 3.1: A virtual character pushing an obstacle according to the operator's order.

work, which enables a virtual character to compromise between following an operator's motions, and adjusting postures by itself to perform tasks in a physics based virtual environment. The virtual character is considered as a mechanical system influenced by multiple wrenches. The motion of each task frame is guided by its task wrench. Multiple tasks are handled by regulating task wrenches.

3.2 Related work

Our control framework is inspired by the work of [Collette 2008] and [Abe 2007], who have both proposed a framework of multi-objective control based on optimization. Their approaches use acceleration-based control, which defines the objective as:

$$\arg\min_{\dot{\mathbf{T}}} \left\| \mathbf{J}\dot{\mathbf{T}} + \dot{\mathbf{J}}\mathbf{T} - \ddot{\mathbf{X}}^d \right\| \tag{3.1}$$

with $\ddot{\mathbf{X}}^d$ the desired acceleration derived from the task to perform. However, such a method may lead to numerical singularities for certain poses, since the process of solving for joint accelerations $\dot{\mathbf{T}}$ in (3.1) boils down to computing the inverse of the Jacobian. Moreover, it is difficult for the character to adapt to sudden environment changes related to new established contacts during manipulation by using acceleration-based control. To deal with these problems, our system uses force-based control as an alternative approach, which makes the interaction more feasible and safe.

Our force-based control framework adopts a **Jacobian-transpose** (JT) control method, which has been used in [Pratt 1996] for virtual actuator control, and in [Coros 2010] for a walking controller. In [Demircan 2008], human motion from motion capture data is reconstructed, by using a simplified version of the framework that is proposed in [Khatib 2008]. This simplified version is close to the JT control method.

The principle of our control framework is similar to those presented in [Wu 2010] and [Stephens 2010], both of which combined the JT method with optimization. The work in [Wu 2010] proposed a static resolution of forces based on the relations of some action-reaction frame pairs. For each frame pair, a force variable applied at the action frame from the reaction frame is defined, as well as an opposite force variable applied at the reaction frame; then, optimization is used to solve for these variables. Compared with such a method, ours is more general in the sense that it is not needed to identify action-reaction frame pairs. Each frame is associated with one force variable; therefore, the number of optimization variables in our framework is the same as the number of task frames; however, if there are many body frames that interact with one another, the variable number becomes much larger with the method in [Wu 2010]. Besides, contact constraints that are associated with friction cones were not considered in [Wu 2010]. In [Stephens 2010], a synthesis of control laws involving two main steps was proposed. The first step consists of estimating ground contact forces, which are used in the second step to compute joint torques by solving dynamics and constraint equations. These equations in the second step are solved by a damped pseudo-inverse, which leads to a solution that minimizes the norm of the vector $\left[\begin{array}{cc} \dot{\mathbf{T}}^T & \boldsymbol{\tau}^T \end{array} \right]^T$. This may, in most situations, have the risk of generating unwanted behaviors or movements; for example, consider a simple gravity compensation, for which we do not want to move the virtual character at all, but the use of a damped pseudo-inverse may lead to $\dot{\mathbf{T}} \neq \mathbf{0}$.

3.3 Dynamics of the virtual character

The dynamics of the virtual character (Fig.3.2) is considered as a second order system (3.2).

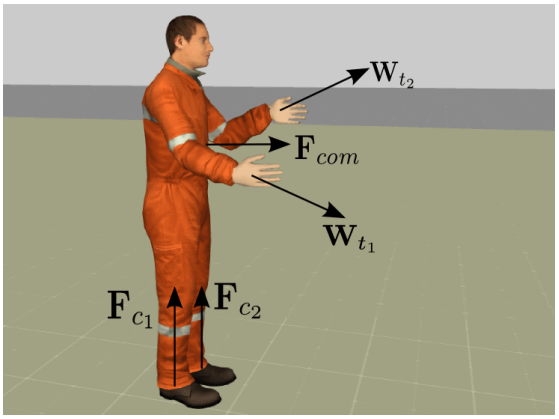


Figure 3.2: Example of a virtual character with wrenches associated with different frames.

$$\mathbf{M}\dot{\mathbf{T}} + \mathbf{N}\mathbf{T} + \boldsymbol{\gamma}^r = \mathbf{L}\boldsymbol{\tau} - \sum_j \mathbf{J}_{c_j}^T \mathbf{W}_{c_j}^r - \sum_k \mathbf{J}_{p_k}^T \mathbf{W}_{p_k}^r \tag{3.2}$$

In the notations of this chapter, frames are denoted by subscripts as follows:

- *com* for CoM frame;
- *c* for non-sliding contacts at fixed locations, which are known a priori, such as the contacts between the feet and the ground;
- *p* for contacts where the environment is not fixed but behaves passively, which means that the virtual character experiences a passive interaction with the environment; these contacts are unknown a priori;
- *t* for task frames that are associated with the end-effector motion control.

Note that only passive interactions between the virtual character and the environment are considered here. The restriction to passive interactions is a natural choice, since the human can be supposed to behave like a passive environment [Krüger 2008], and a great number of environments with which people and robots interact are passive [Colgate 1989].

Moreover the following superscripts are used:

- *d* for "desired" wrench values for the controller,
- *r* for "real" wrench values in a simulation,
- *root* for the root DoF,
- *ac* for the actuated DoF.

With these notations, a Jacobian matrix $\mathbf{J}$ is decomposed as $\mathbf{J} = \begin{bmatrix} \mathbf{J}^{root} & \mathbf{J}^{ac} \end{bmatrix}$.

3.4 Control framework

The control framework is shown in Fig. 3.3. At each time step of simulation, the control system computes joint torques from a motion capture sequence, given tasks, and constraints. The whole control is divided into two steps: the first step is the computation of the optimal wrenches; joint torques are then computed according to the optimal wrenches in the second step. The joint torques are used to drive the virtual character.

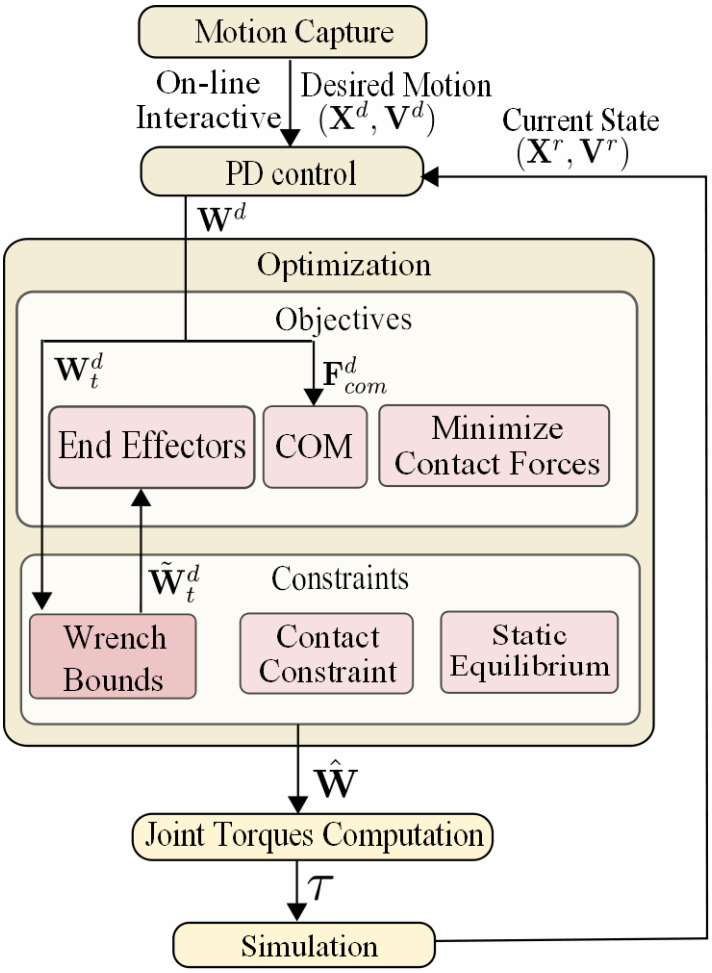


Figure 3.3: Block diagram of the control framework.

3.4.1 Virtual wrenches computation

The principle of the JT control method [Demircan 2008,Pratt 1996] is as follows. Given a task wrench $\mathbf{W}$ in the space of Cartesian coordinates, the equivalent joint torques $\boldsymbol{\tau}$ can be obtained by $\boldsymbol{\tau} = \mathbf{J}^T\mathbf{W}$, with $\mathbf{J}$ the Jacobian matrix at the point where $\mathbf{W}$ is supposed to be applied.

In our multi-objective control, a task means that a certain frame on the virtual character's body should be transferred from an initial state to a desired state. For each task, imagine that a "virtual" wrench is applied at a certain frame on the virtual character's body to guide its motion toward a given target. These virtual wrenches are defined to be applied by virtual characters on environments. They are computed by solving an optimization problem.

The optimization variables are the CoM task force $\mathbf{F}_{com}$, the end-effector task wrenches $\mathbf{W}_t$, and the contact forces $\mathbf{F}_c$. The optimization tries to find optimal task wrenches by taking into account their desired values, as well as the constraints of static equilibrium, non-sliding contacts, and wrench bounds.

Suppose there are n task frames to control and m fixed contacts. The optimization problem is written as follows:

$$\underset{\mathbf{F}_{com},\mathbf{W}_{t_i},\mathbf{F}_{c_j}}{\arg\min} \quad \frac{1}{2}\left\| \begin{bmatrix} \mathbf{F}^d_{com} \\ \mathbf{W}^d_{t_i} \\ \mathbf{F}^d_{c_j} \end{bmatrix} - \begin{bmatrix} \mathbf{F}_{com} \\ \mathbf{W}_{t_i} \\ \mathbf{F}_{c_j} \end{bmatrix} \right\|^2_{\mathbf{Q}} \tag{3.3a}$$

$$\text{subject to} \quad \mathbf{J}^{root^T}_{com}\mathbf{F}_{com} + \textstyle\sum_i \mathbf{J}^{root^T}_{t_i}\mathbf{W}_{t_i} + \textstyle\sum_j \mathbf{J}^{root^T}_{c_j}\mathbf{F}_{c_j} + \boldsymbol{\gamma}^{r,root} = 0 \tag{3.3b}$$

$$\mathbf{A}_{c_j}\mathbf{F}_{c_j} - \mathbf{d}_{c_j} > 0 \tag{3.3c}$$

$$\mathbf{W}^{min}_t \leq \mathbf{W}_t \leq \mathbf{W}^{max}_t \tag{3.3d}$$

with $i = 1, 2, ..., n$ and $j = 1, 2, ..., m$. The optimization objective is the same for each task, which is to minimize the error between the variable and its desired value. The objectives are combined by the diagonal weight matrix $\mathbf{Q}$, whose value is chosen according to the importance levels or the priorities of different objectives.

3.4.1.1 Target tracking objectives

The two primary kinds of target tracking tasks considered here are the CoM task and the end-effector tasks. Our control system takes the CoM as the stability criterion and maintains balance by controlling its position. The end-effector tasks can be either tracking a captured motion sequence or performing some specific motions. For each task, the desired task wrench $\mathbf{W}^d$ is computed by using a proportional-derivative (PD) feedback control law.

$$\mathbf{W}^d = \mathbf{K}\delta(\mathbf{H}^d, \mathbf{H}^r) + \mathbf{B}\delta(\mathbf{V}^d, \mathbf{V}^r) \tag{3.4}$$

with $\mathbf{H}^r \in \mathbf{SE(3)}$, $\mathbf{H}^d \in \mathbf{SE(3)}$, $\mathbf{V}^r \in \mathbf{se(3)}$ and $\mathbf{V}^d \in \mathbf{se(3)}$, where $\mathbf{SE(3)}$ is the special Euclidean group and $\mathbf{se(3)}$ is the Lie algebra of $\mathbf{SE(3)}$. $\delta(\mathbf{H}^d, \mathbf{H}^r)$ denotes the displacement (position and orientation) error between the desired and current states, while $\delta(\mathbf{V}^d, \mathbf{V}^r)$ denotes the velocity (linear and angular velocity) error between the desired and current states. $\mathbf{K}$ and $\mathbf{B}$ denote the proportional and derivative gain matrices respectively.

For the end-effector tasks, both the position and orientation errors are considered; for the CoM task, only the position error is considered.

If a contact is established between the end effector and an object during manipulation, an additional offset value can be added to the result of (3.4).

$$\mathbf{W}^d = \mathbf{K}\delta(\mathbf{H}^d, \mathbf{H}^r) + \mathbf{B}\delta(\mathbf{V}^d, \mathbf{V}^r) + \mathbf{W}^{\text{offset}} \tag{3.5}$$

This offset $\mathbf{W}^{\text{offset}}$ helps to enhance the contact between the end effector and the object.

3.4.1.2 Contact force objective

For each fixed contact, only the force component $\mathbf{F}_{c_j}$ of the wrench $\mathbf{W}_{c_j}$ is considered. The contact force objective is not a target tracking objective. It is used here for numerical reason, so that the matrix $\mathbf{Q}$ is full rank. As we do not need to know the appropriate value of each contact force *a priori*, $\mathbf{F}^d_{c_j}$ is set to zero, and a weak weight is assigned to this objective. Then we let the optimization find the appropriate contact forces.

3.4.1.3 Static equilibrium constraint

The wrenches are constrained by the static equilibrium of the root body under $\mathbf{F}_{com}$, $\mathbf{W}_{t_i}$, $\mathbf{F}_{c_j}$, and $\boldsymbol{\gamma}^r$. This constraint is written in (3.3b).

3.4.1.4 Contact constraints

The contact force objective mentioned in 3.4.1.2 only tries to minimize the contact forces $\mathbf{F}_{c_j}$, more appropriate values of which are computed by the optimization. The optimization searches for a contact force which not only satisfies the static equilibrium in (3.3b), but remains inside the friction cone as well, in order to maintain a non-sliding contact. The linearized Coulomb friction model [Abe 2007,Collette 2007] is applied, in which the friction cone of each contact is approximated by a four-faced polyhedral convex cone. The contact constraints are written in (3.3c), with

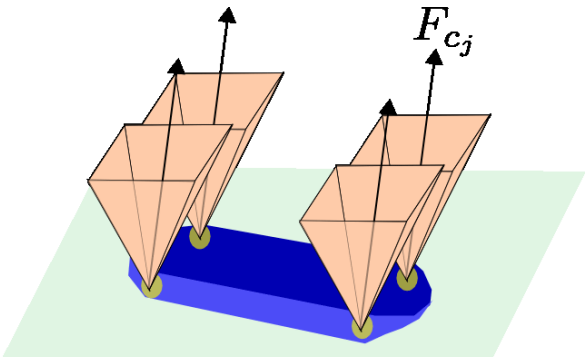


Figure 3.4: Friction cone.

$$\mathbf{A}_{c_j} = \begin{bmatrix} \lambda_1 \times \lambda_2 & \lambda_2 \times \lambda_3 & \lambda_3 \times \lambda_4 & \lambda_4 \times \lambda_1 \end{bmatrix}^T, \tag{3.6}$$

where λ is the unit edge vector of the approximated friction cone, and $\mathbf{d}_{c_j}$ is a user defined margin vector. The projection of $\mathbf{F}_{c_j}$ on the normal vector of each facet of the friction cone is constrained to be larger than $\mathbf{d}_{c_j}$.

3.4.1.5 Wrench bounds

Wrenches of lower-priority tasks are bounded if these tasks conflict with a higher-priority task. Here, wrench bounds associated with motions in SE(3) are considered. The

computation of wrench bounds is explained in detail in Chapter 4.

3.4.2 Joint torques computation

Joint torques are computed in (3.7) using the solution ($\hat{\mathbf{F}}_{com}$, $\hat{\mathbf{W}}_{t_i}$, and $\hat{\mathbf{F}}_{c_j}$) of the optimization.

$$\boldsymbol{\tau} = \mathbf{J}_{com}^{ac\ T}\hat{\mathbf{F}}_{com} + \sum_i \mathbf{J}_{t_i}^{acT}\hat{\mathbf{W}}_{t_i} + \sum_j \mathbf{J}_{c_j}^{acT}\hat{\mathbf{F}}_{c_j} + \boldsymbol{\gamma}^{r,ac}, \tag{3.7}$$

3.5 Results

The proposed control framework has been implemented on the virtual character described in Section 1.2. Experiments have been conducted to demonstrate that the proposed control system is capable of driving the virtual character to perform a desired manipulation task in an interactive manner, and to change its posture automatically so as to handle unknown external contact forces.

We have applied our control system to the following object manipulation tasks, all of which are performed by interactions with the operator:

- Push storage cabinets (Fig. 3.5).
- Take a box up from a table, then move it (Fig. 3.6).

The storage cabinets weigh up to $50kg$. The box weighs $10kg$. The friction coefficients between the feet and the ground, as well as between the objects and the hands are 1.0. The objects to manipulate only exist in the virtual environment. The CoM task is used for balance control. The hand tasks are used for reaching or manipulation control. The control has been realized in real-time with a simulation time step of $0.01s$.

The operator wears motion tracking markers on his hands. Only the motions of the operator's hands are captured. During real-time interactions, the virtual character is animated according to the hand displacements of the operator, which are taken as the reference hand displacements for the virtual character. Once the virtual character's hands are in contact with the object, it starts to perform manipulation according to the guidance of the operator's motions. The whole-body motions of the virtual character are generated by our multi-objective controller.

The virtual character's actions are determined based on events from the operator and

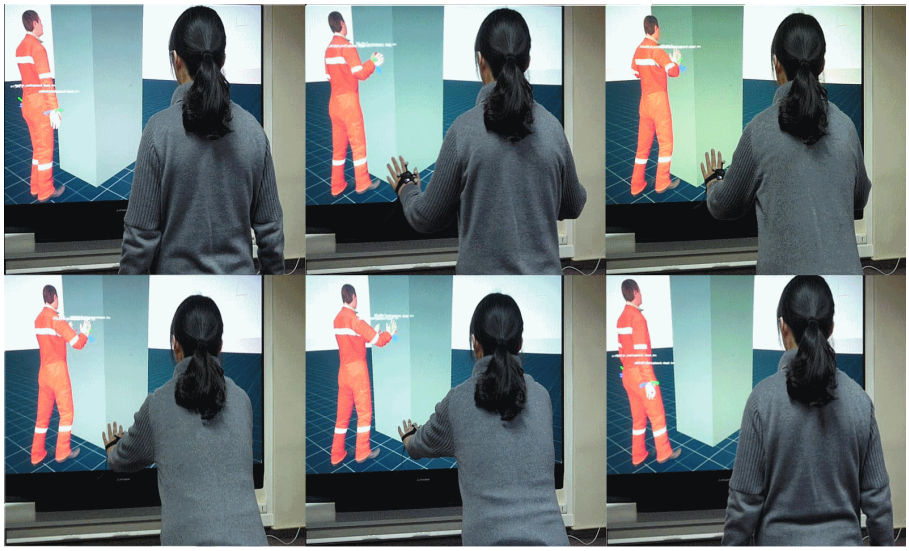


Figure 3.5: Snapshots of the virtual character pushing a storage cabinet according to interactions with the operator.

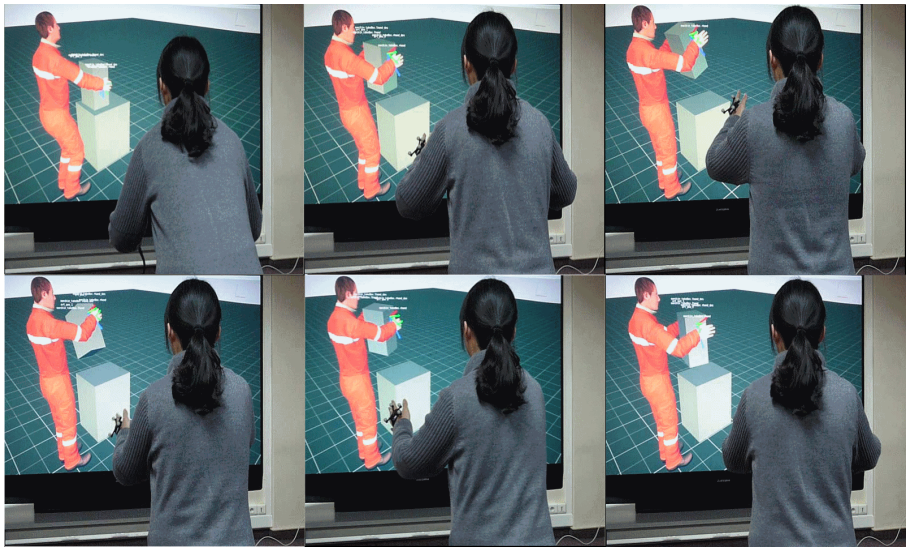


Figure 3.6: Snapshots of the character taking a box up, moving it, and then putting it down.

the simulation environment. Currently, the actions of the virtual character are fixed for a given task. The virtual character decides how to react according to a finite state machine (FSM) illustrated in Fig.3.7. In the future, we plan to develop a more generic method for intention detection. Hereby, the goal of the experiments is to assess the validity of our control framework, as well as to illustrate the problematic of our future works on the topic.

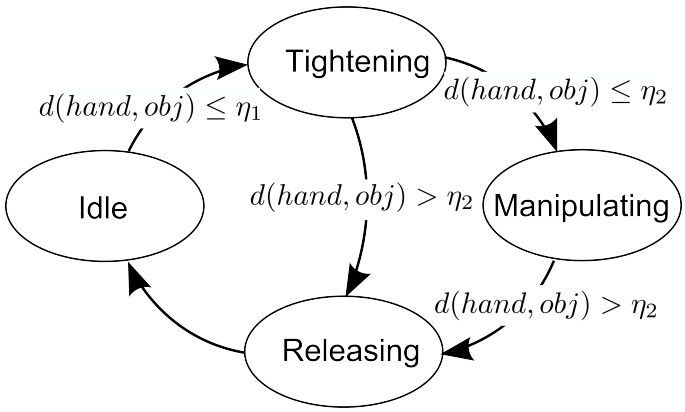


Figure 3.7: A finite state machine for action decision. d(hand, obj) denotes the distance between the operator's hand position expressed in the virtual environment and the position of the object. η_1 and η_2 are two threshold values with $\eta_2 > \eta_1$.

- **Idle state**: At the beginning of the simulation, the character is in the Idle state. In this state, the character is not in contact with the object to manipulate, and his hands follow the motions of the operator's hands.
- **Tightening state**: If the distance between the operator's hand position expressed in the virtual environment and the position of the object d(hand, obj) is smaller than a threshold value η_1, then the character's hands should make contact with the object. At this moment, the operator's hands may continue moving; but the character cannot simply follow the operator's motions. Instead, it has to deal with the interactions between itself and the object in the virtual environment. During the Tightening state, the contact forces between the character's hands and the object increase gradually until the character can move the object.
- **Manipulating state**: In this state, the character's hands are in contact with the object. At the same time, the character tries to move the object according to the

motions of the operator's hands.

– **Releasing state**: The operator sends out an interruption order for stopping task execution by moving his hands away. Once the value of $d(\text{hand}, \text{obj})$ is larger than another threshold value η_2 ($\eta_2 > \eta_1$), the character starts to release the object by gradually reducing the contact forces. Once the action of releasing is performed, the character returns to the Idle state.

As mentioned in Section 3.4.1.1, the contact forces during manipulation can be adjusted by our control approach. Once contacts are established between the hands and an object, an additional offset value $\mathbf{W}_{\text{hand}}^{\text{offset}}$ is added to the desired hand task wrenches to enhance the contact. This offset value can be defined according to the physical characteristic of the object, such as the weight and the friction coefficient. When moving an object, the hand contact force offset $\mathbf{W}_{\text{hand}}^{\text{offset}}$ is set to be perpendicular to the contact surface during manipulation. In this way, the character is able to strengthen and maintain the contacts with the object, so as to avoid from dropping it during the manipulation. By simply emulating captured motions, it is possible to manipulate an object which is not too heavy; however, to be able to take up heavier objects, the operator has to move his hands much closer to each other, so as to generate stronger contact forces with the object. As there is a limitation of how close the operator's hands can be to each other, this can result in a limit of the weight of the object which the character is able to carry. So an advantage of our approach over some motion editing approaches, which try to simply emulate captured motions, is that desired contact forces can be directly taken into account in our controller.

3.6 Discussion

3.6.1 Shared control

The experiments suggest that the proposed control framework can successfully control the virtual character to perform manipulation tasks through interaction with an operator. During human-virtual character interaction, the computer and the human operator share their control on the virtual character. The operator sends higher-level task indications; the controller handles these tasks from a low level to respect the operator's intentions as much as possible. Such a mode is sometimes referred to as shared control or human-in-

the-loop [Chipalkatty 2010].

In our shared control, the operator may not be able to provide an appropriate posture for a manipulation task, since the operator cannot really apply forces on a virtual object; however, by taking account of all the objectives and constraints in the optimization, the controller can adjust captured motions to handle interactions with the virtual environment. Moreover, the wrench-bound constraints mentioned in Section 3.4.1.5 further ensures the balance of the virtual character by assigning the CoM task with a higher priority than all the other motion tracking tasks. As a result, the virtual character can move the end-effector in its reachable space safely, and try its best to track the captured motions. An advantage of our approach is that the operator can provide any reference motion to guide the virtual character, without being worried about the virtual character's balance problem.

3.6.2 The choice of optimization objective weights

In our experiments, the weight of contact force objective is set to 1. The weights that are associated with target tracking objectives can simply be chosen according to their importance levels. The least important one is assigned with a weight of 100. If the importance of task i is one level higher than the importance of task j, then the weight of task i is increased to ten times the weight of task j.

3.7 Conclusions

A novel control framework for virtual characters, which combines multi-objective control with motion capture techniques, has been presented. The wrench-based control policy allows the virtual character to be adapted to unknown external contact forces and handle unilateral constraints on contact forces; therefore, the controller can generate motions suitable in a virtual environment, rather than simply emulating captured motions.

Our approach is suitable for interactive applications such as virtual reality. It can work fast enough to allow the virtual character to interact with an operator and the virtual environment in real-time. The effectiveness of the proposed approach has been demonstrated by experiments.

Chapter 4

A Two-Level Prioritized Control with Wrench Bounds

Contents

This chapter extends Chapter 3 by describing in detail a method to compute wrench bounds for handling conflicting tasks. Bounds are imposed on lower-priority task wrenches to ensure the controller performance of higher-priority tasks. These wrench bounds are used in our multi-objective control framework to compute optimal task wrenches. The novelty of this method is that it can handle inequality constraints on a higher-priority task and maintain passivity as well. This method allows an operator to interact with the

virtual character in real-time, without the necessity of compromising the virtual character's balance. Simulations demonstrate that it can improve the behavior of a virtual character.

4.1 Introduction

A virtual character is often required to perform multiple tasks simultaneously. For example, we can define a task for the CoM for balance control, or tasks for the end-effectors for motion tracking control. While multiple tasks are performed simultaneously, some tasks can be incompatible with one or another. The main problem we focus on in this chapter is how to handle conflicts among multiple tasks.

The contribution of this chapter is the development of a prioritized control approach based on **wrench bounds**. Wrench bounds are imposed on lower-priority task wrenches to ensure that they will not drive a higher-priority task frame out of its admissible domain. This method allows inequality constraints on a higher-priority task, and takes into consideration the passivity of the system.

4.2 Related work

A classical method to realize prioritized control is by using **null space projections** [Liégeois 1977,Sentis 2004,Khatib 2008,Yoshida 2010], in which a lower-priority task is satisfied only in the null space of higher-priority tasks. The null space projection ensures that lower-priority tasks are controlled without dynamically interfering with higher-priority tasks [Khatib 2004,Raunhardt 2011].

Inequality constraints on lower-priority tasks can be realized by the projection of an artificial potential field term [Khatib 1986] onto the null space of equality constraints, which proved especially efficient for task objectives such as joint limits or object avoidance [Sentis 2005,Stasse 2008]. A repulsive potential is used in [Saab 2009] to keep the projected center of mass (CoM) away from the boundary of the support polygon, and the gradient of the potential is projected onto the null space of the hand task. But in this case, the CoM task is of lower-priority, so its controller performance can no longer be ensured. Moreover, when applying such a prioritization method, the higher-priority task frame will remain on

its desired position, which can be too restrictive and may either reduce the workspace or result in more body movements in order to fulfill lower-priority tasks.

Instead of being constrained to a fixed position, a higher-priory task frame is sometimes allowed to move with a margin of error, within which the task performance can be ensured. Therefore, a prioritization method which allows inequality constraints at a higher-priority level is needed. Null space projector has been adopted in [Mansard 2009] to integrate unilateral constraints at any priority level. This approach has achieved impressive results, although the computation of some specific inverse operators is complex and time consuming. An alternative solution is to use **a sequence of quadratic programs** [Kanoun 2009,Escande 2010,Saab 2011,Kanoun 2011a]. Such kind of prioritization process boils down to the classical algorithm based on null space projections when only linear equalities are considered [Kanoun 2009]. All the methods mentioned here, among many others, rely on null space projections; but it is shown in [Rennuit 2005,Rennuit 2006] that prioritization based on projections can **break passivity**, which is a sufficient condition to guarantee stable operations [Hannaford 2002,Kim 2010].

To avoid breaking passivity, we propose to impose wrench bounds on lower-priority tasks, so as to guarantee that they are fulfilled only if their task wrenches will not drive a higher-priority task frame out of its admissible domain. This prioritized control can be adaptable to real-time motion tracking tasks. Wrench bounds are computed automatically at each time step. There is no need to manually tune control parameters, such as weights or gains used in the optimization. Inequality constraints on a higher-priority task are allowed, which means the higher-priority task frame is allowed to move as long as the error remains within a tolerance margin. This provides lower-priority tasks with more freedom of movement, since they are to some extent allowed to dynamically interfere with a higher-priority task. The energy is bounded in our method; therefore, the system is passive, since it cannot supply power indefinitely.

4.3 Wrench-bound computation

Our two-level prioritized control is realized by imposing bounds on lower-priority task wrenches to guarantee that a higher-priority task can be fulfilled.

To explain the idea of this method, suppose that there are w virtual wrenches, one of which is associated with a higher-priority task. Let $\mathcal{L}^+ = \{0, 1, 2, ..., w-1\}$ and $\mathcal{L} = \mathcal{L}^+ \backslash \{0\}$. The w wrenches are denoted as $\{\mathbf{W}_l : l \in \mathcal{L}^+\}$. The wrench which is associated with the higher-priority task is denoted as $\mathbf{W}_0$. Each lower-priority task wrench is denoted as $\mathbf{W}_l$ with $l \in \mathcal{L}$.

4.3.1 Preliminary conditions

This wrench-bound computation is based on the following conditions:

- Task targets are constant during each time step.
- An admissible domain exists for a higher-priority frame.
- The sum of the kinetic and potential energy at the initial time t_0 should be no larger than a threshold value $\mathbf{U}_0^{max}$, which will be defined later in this chapter.

The admissible domain of a frame represents a tolerance margin around a desired configuration. The performance of a higher-priority task is ensured by constraining the task frame inside its admissible domain. For example, if the virtual character is standing on the horizontal ground, the admissible domain of the CoM should be defined in such a way that its vertical projection is inside the support polygon. The CoM should always lie inside its admissible domain to maintain balance. The bounds of lower-priority task wrenches should be found, such that the higher-priority frame is kept within its admissible domain.

4.3.2 Elastic potential energy associated with a wrench

The potential energy which is associated with a target position $\mathbf{X}_l^d$ has been used in [Liu 2011a] for the computation of constraints on translational movement. Such a method can be extended by defining a potential energy function $\mathbf{U}_l$ for each wrench $\mathbf{W}_l$ associated with a desired displacement $\mathbf{H}_l^d$. This potential energy $\mathbf{U}_l$ will be used later for the computation of wrench bounds.

The potential energy as a function of a body configuration $\mathbf{H}$ has been studied in [Fasse 1998], where the potential energy function is decomposed into translational, rotational, and coupling terms. The total potential energy is the sum of these terms. The

coupling term is not considered here; therefore, the total potential energy for $\mathbf{W}_l$ is defined as

$$\mathbf{U}_l(\mathbf{H}_l^d, \mathbf{H}_l) = \mathbf{U}_l^{\text{tr}}(\mathbf{X}_l^d, \mathbf{X}_l) + \mathbf{U}_l^{\text{rot}}(\mathbf{R}_l^d, \mathbf{R}_l), \tag{4.1}$$

where the translational potential energy $\mathbf{U}_l^{\text{tr}}$ is defined as

$$\mathbf{U}_l^{\text{tr}}(\mathbf{X}_l^d, \mathbf{X}_l) = \frac{1}{2}(\mathbf{X}_l^d - \mathbf{X}_l)^T \mathbf{K}_l^{\text{tr}}(\mathbf{X}_l^d - \mathbf{X}_l), \tag{4.2}$$

and the rotational potential energy $\mathbf{U}_l^{\text{rot}}$ is defined as

$$\mathbf{U}_l^{\text{rot}}(\mathbf{R}_l^d, \mathbf{R}_l) = tr(\mathbf{G}_l^{\text{rot}}) - tr(\mathbf{G}_l^{\text{rot}} \mathbf{R}_l^{-1} \mathbf{R}_l^d) \tag{4.3}$$

$$\mathbf{G}_l^{\text{rot}} = \frac{1}{2} tr(\mathbf{K}_l^{\text{rot}}) \mathbf{I} - \mathbf{K}_l^{\text{rot}} \tag{4.4}$$

with $\mathbf{K}_l^{\text{tr}}$ and $\mathbf{K}_l^{\text{rot}}$ being the translational and the rotational part of the stiffness matrix $\mathbf{K}_l$, respectively, and $\mathbf{G}_l^{\text{rot}}$ being the co-stiffness matrix [Fasse 1998] which is associated with the stiffness matrix $\mathbf{K}_l^{\text{rot}}$. The term $tr(\mathbf{G}_l^{\text{rot}})$ is added to the form of the rotational potential energy defined in [Fasse 1998] to ensure the non-negative property of $\mathbf{U}_l^{\text{rot}}$ (see Appendix A.1). With the forms defined in (4.2) and (4.3), both $\mathbf{U}_l^{\text{tr}}$ and $\mathbf{U}_l^{\text{rot}}$ are non-negative; thus, $\mathbf{U}_l(\mathbf{H}_l^d, \mathbf{H}_l)$ is non-negative.

In order to bound wrenches, a scale variable α_l is added into the potential energy function of each lower-priority task:

$$\mathbb{U}_l(\alpha_l, \mathbf{H}_l^d, \mathbf{H}_l) = \alpha_l \mathbf{U}_l(\mathbf{H}_l^d, \mathbf{H}_l),\ l \in \mathcal{L}. \tag{4.5}$$

For the sake of simplicity, $\mathbf{U}_l(\mathbf{H}_l^d, \mathbf{H}_l)$ will be denoted as $\mathbf{U}_l$ in the rest of this chapter. We use $\mathbf{U}_0^{max}$ to denote the maximum allowable value of $\mathbf{U}_0$. When $\mathbf{H}_0$ moves to the edge of its admissible domain, $\mathbf{U}_0$ increases to this maximum allowable value.

The wrenches are associated with the potential:

$$\mathbf{W}_l = \begin{cases} -\nabla_{\mathbf{H}_l} \mathbf{U}_l - \mathbf{B}_l \mathbf{V}_l,\ l = 0, \\ -\alpha_l \nabla_{\mathbf{H}_l} \mathbf{U}_l - \mathbf{B}_l \mathbf{V}_l,\ l \in \mathcal{L}, \end{cases} \tag{4.6}$$

with $\mathbf{B}_l$ denoting the damping matrix, and the desired velocity $\mathbf{V}_l^d$ being set to zero. Here the operator of gradient is used to denote the differential of the potential energy function with respect to displacements in SE(3). The details of its computation can be found in [Fasse 1998].

4.3.3 Bounds of lower-priority task wrenches

A virtual character is an under-actuated system, which needs to use contact forces to balance the wrenches due to task execution. Therefore, the wrench $\mathbf{W}_l$ should satisfy the following condition.

$\exists$ *contact forces* $\hat{\mathbf{F}}_{c_j}$: (i) *the static equilibrium* (4.7) *is satisfied;* (ii) *the non-sliding contact constraint* (3.3c) *is satisfied:*

$$\begin{cases} \mathbf{0} = \sum_l \mathbf{J}_l^{root^T} \mathbf{W}_l + \sum_j \mathbf{J}_{c_j}^{root^T} \hat{\mathbf{F}}_{c_j} + \boldsymbol{\gamma}^{r,root} & \text{(4.7a)} \\ \boldsymbol{\tau} = \sum_l \mathbf{J}_l^{acT} \mathbf{W}_l + \sum_j \mathbf{J}_{c_j}^{acT} \hat{\mathbf{F}}_{c_j} + \boldsymbol{\gamma}^{r,ac}. & \text{(4.7b)} \end{cases}$$

According to our experiences, the constraints (4.7) and (3.3c) can be satisfied for many tasks which do not require very fast motions. One possible reason for this is that task errors during each time step are small for slow motions, making the values of desired task wrenches not very large; therefore, the tangential contact forces that are required to balance task wrenches will not need to be very large either. Therefore, we make the assumption that the wrench $\mathbf{W}_l$ which is given by this method based on wrench bounds satisfies both (4.7) and (3.3c). Results of our experiments suggest that our method based on this assumption works well. An investigation about the objective measure to assure slowness can be found in Appendix A.2.

Using the right members of (4.7a) and (4.7b) in the dynamics of the system (3.2) leads to

$$\begin{aligned} \mathbf{M}\dot{\mathbf{T}} + \mathbf{N}\mathbf{T} &= \begin{bmatrix} \sum_l \mathbf{J}_l^{root^T} \mathbf{W}_l + \sum_j \mathbf{J}_{c_j}^{root^T} \hat{\mathbf{F}}_{c_j} + \boldsymbol{\gamma}^{r,root} \\ \sum_l \mathbf{J}_l^{acT} \mathbf{W}_l + \sum_j \mathbf{J}_{c_j}^{acT} \hat{\mathbf{F}}_{c_j} + \boldsymbol{\gamma}^{r,ac} \end{bmatrix} \\ &\quad - \sum_j \mathbf{J}_{c_j}^T \mathbf{F}_{c_j}^r - \sum_k \mathbf{J}_{p_k}^T \mathbf{W}_{p_k}^r - \boldsymbol{\gamma}^r \\ &= \sum_l \mathbf{J}_l^T \mathbf{W}_l - \sum_k \mathbf{J}_{p_k}^T \mathbf{W}_{p_k}^r + \sum_j \mathbf{J}_{c_j}^T (\hat{\mathbf{F}}_{c_j} - \mathbf{F}_{c_j}^r). \end{aligned} \tag{4.8}$$

Separating the higher-priority task 0 from the others yields

$$\mathbf{M}\dot{\mathbf{T}} + \mathbf{N}\mathbf{T} = \mathbf{J}_0^T \mathbf{W}_0 + \sum_{l \in \mathcal{L}} \mathbf{J}_l^T \mathbf{W}_l - \sum_k \mathbf{J}_{p_k}^T \mathbf{W}_{p_k}^r + \sum_j \mathbf{J}_{c_j}^T (\hat{\mathbf{F}}_{c_j} - \mathbf{F}_{c_j}^r). \tag{4.9}$$

Substituting the expression of $\mathbf{W}_l$ (4.6) into (4.9) leads to

$$\begin{aligned}\mathbf{M}\dot{\mathbf{T}} + \mathbf{N}\mathbf{T} = &-\mathbf{J}_0^T \nabla_{\mathbf{H}_0} \mathbf{U}_0 - \sum_{l\in\mathcal{L}} \alpha_l \mathbf{J}_l^T \nabla_{\mathbf{H}_l} \mathbf{U}_l \\ &- (\mathbf{J}_0^T \mathbf{B}_0 \mathbf{J}_0 + \sum_{l\in\mathcal{L}} \mathbf{J}_l^T \mathbf{B}_l \mathbf{J}_l)\mathbf{T} \\ &- \sum_k \mathbf{J}_{p_k}^T \mathbf{W}_{p_k}^r + \sum_j \mathbf{J}_{c_j}^T (\hat{\mathbf{F}}_{c_j} - \mathbf{F}_{c_j}^r).\end{aligned} \tag{4.10}$$

Multiplying both sides of (4.10) with $-\mathbf{T}^T$ yields

$$\begin{aligned}&-\mathbf{T}^T\mathbf{M}\dot{\mathbf{T}} - \mathbf{T}^T\mathbf{N}\mathbf{T} - \mathbf{T}^T(\mathbf{J}_0^T\mathbf{B}_0\mathbf{J}_0 + \sum_{l\in\mathcal{L}} \mathbf{J}_l^T\mathbf{B}_l\mathbf{J}_l)\mathbf{T} \\ =&\mathbf{V}_0^T \nabla_{\mathbf{H}_0}\mathbf{U}_0 + \sum_{l\in\mathcal{L}} \alpha_l \mathbf{V}_l^T \nabla_{\mathbf{H}_l}\mathbf{U}_l \\ &+ \sum_k \mathbf{V}_{p_k}^T \mathbf{W}_{p_k}^r - \sum_j \mathbf{v}_{c_j}^T (\hat{\mathbf{F}}_{c_j} - \mathbf{F}_{c_j}^r) \\ =&\frac{d\mathbf{U}_0}{dt} - {\mathbf{V}_0}^{dT} \nabla_{\mathbf{H}_0^d}\mathbf{U}_0 \\ &+ \sum_{l\in\mathcal{L}} (\frac{d(\alpha_l\mathbf{U}_l)}{dt} - \dot{\alpha}_l\mathbf{U}_l - \alpha_l {\mathbf{V}_l}^{dT} \nabla_{\mathbf{H}_l^d}\mathbf{U}_l) \\ &+ \sum_k \mathbf{V}_{p_k}^T \mathbf{W}_{p_k}^r - \sum_j \mathbf{v}_{c_j}^T (\hat{\mathbf{F}}_{c_j} - \mathbf{F}_{c_j}^r).\end{aligned} \tag{4.11}$$

For non-sliding contacts with a fixed environment, the velocity $\mathbf{v}_{c_j} = 0$; thus, we have

$$\begin{aligned}&-\mathbf{T}^T\mathbf{M}\dot{\mathbf{T}} - \mathbf{T}^T\mathbf{N}\mathbf{T} - \mathbf{T}^T(\mathbf{J}_0^T\mathbf{B}_0\mathbf{J}_0 + \sum_{l\in\mathcal{L}} \mathbf{J}_l^T\mathbf{B}_l\mathbf{J}_l)\mathbf{T} \\ =&\frac{d\mathbf{U}_0}{dt} - {\mathbf{V}_0}^{dT} \nabla_{\mathbf{H}_0^d}\mathbf{U}_0 \\ &+ \sum_{l\in\mathcal{L}} (\frac{d(\alpha_l\mathbf{U}_l)}{dt} - \dot{\alpha}_l\mathbf{U}_l - \alpha_l {\mathbf{V}_l}^{dT} \nabla_{\mathbf{H}_l^d}\mathbf{U}_l) \\ &+ \sum_k \mathbf{V}_{p_k}^T \mathbf{W}_{p_k}^r.\end{aligned} \tag{4.12}$$

Integrating (4.12) with respect to time t from t_0 to $t_0 + T$ yields

$$\begin{aligned}
&\mathbf{E}^{t_0} - \mathbf{E}^{t_0+T} - \int_{t_0}^{t_0+T} \mathbf{D}^t dt \\
=&\mathbf{U}_0^{t_0+T} - \mathbf{U}_0^{t_0} - \int_{t_0}^{t_0+T} \mathbf{V}_0{}^{d,t\,T} \nabla_{\mathbf{H}_0^{d,t}} \mathbf{U}_0^t dt \\
&+ \sum_{l\in\mathcal{L}} (\alpha_l^{t_0+T} \mathbf{U}_l^{t_0+T} - \alpha_l^{t_0} \mathbf{U}_l^{t_0}) \\
&- \int_{t_0}^{t_0+T} \sum_{l\in\mathcal{L}} (\dot{\alpha}_l{}^t \mathbf{U}_l^t + \alpha_l^t \mathbf{V}_l{}^{d,t\,T} \nabla_{\mathbf{H}_l^{d,t}} \mathbf{U}_l^t) dt \\
&+ \int_{t_0}^{t_0+T} \sum_k \mathbf{V}_{p_k}^t{}^T \mathbf{W}_{p_k}^{r,t} dt,
\end{aligned} \tag{4.13}$$

with

$$\begin{aligned}
\mathbf{D}^t &= \mathbf{T}^{t\,T} (\mathbf{J}_0^t{}^T \mathbf{B}_0 \mathbf{J}_0^t + \sum_{l\in\mathcal{L}} \mathbf{J}_l^t{}^T \mathbf{B}_l \mathbf{J}_l^t) \mathbf{T}^t, \\
\mathbf{E}^t &= \int_{t_0}^{t} (\mathbf{T}^{t\,T} \mathbf{M}^t \dot{\mathbf{T}}^t + \mathbf{T}^{t\,T} \mathbf{N}^t \mathbf{T}^t) dt,
\end{aligned} \tag{4.14}$$

where $\mathbf{D}$ is for dissipation, and the superscript t indicates the value of a term at time t. By using integration by parts, and noting that $\dot{\mathbf{M}} - 2\mathbf{N}$ is skew-symmetric [Spong 2005], the expression of $\mathbf{E}^t$ in (4.14) gives the kinetic energy at time t.

Let $\mathbf{\Phi}$ denote the sum of the kinetic and potential energy. According to [Hannaford 2002], the term $\int_{t_0}^{t_0+T} \sum_k \mathbf{V}_{p_k}^t{}^T \mathbf{W}_{p_k}^{r,t} dt$ should be non-negative for a passive interaction. Applying this property in (4.13) yields

$$\mathbf{\Phi}^{t_0+T} \leq \mathbf{\Phi}^{t_0} + \mathbf{\Lambda}^{t_0+T}, \tag{4.15}$$

with

$$\begin{aligned}
\mathbf{\Phi}^t =& \mathbf{E}^t + \mathbf{U}_0^t + \sum_{l\in\mathcal{L}} \alpha_l^t \mathbf{U}_l^t, \\
\mathbf{\Lambda}^t =& \int_{t_0}^{t} \dot{\mathbf{\Lambda}}^t dt, \\
\dot{\mathbf{\Lambda}}^t =& \sum_{l\in\mathcal{L}} (\dot{\alpha}_l^t \mathbf{U}_l^t + \alpha_l^t \mathbf{V}_l^{d,t\,T} \nabla_{\mathbf{H}_l^{d,t}} \mathbf{U}_l^t) + \mathbf{V}_0^{d,t\,T} \nabla_{\mathbf{H}_0^{d,t}} \mathbf{U}_0^t - \mathbf{D}^t.
\end{aligned} \tag{4.16}$$

In order to maintain the higher-priority frame inside its admissible domain, the total energy $\mathbf{\Phi}^t$ is constrained to be no larger than $\mathbf{U}_0^{max}$, the maximum allowed value of $\mathbf{U}_0$:

$$\mathbf{\Phi}^t \leq \mathbf{U}_0^{max}, \ \forall t \in [t_0, t_0 + T]. \tag{4.17}$$

Moreover, the potential energy should be non-negative; therefore, α_l should be non-negative. These constraints lead to the following constraint for each α_l at initial time:

$$0 \leq \alpha_l^{t_0} \leq \frac{\mathbf{U}_0^{max} - \mathbf{U}_0^{t_0} - \mathbf{E}^{t_0}}{\sum_{l\in\mathcal{L}} \mathbf{U}_l^{t_0}}, \tag{4.18}$$

which is a sufficient condition for (4.17). Note that according to the third preliminary condition mentioned in Section 4.3.1, we have $\mathbf{\Phi}^{t_0} \leq \mathbf{U}_0^{max}$, which ensures that the upper bound of $\alpha_l^{t_0}$ in (4.18) is non-negative.

Once we have $\alpha_l^{t_0}$ which satisfies (4.18), we now try to increase its value during the simulation, as long as (4.17) is satisfied. To realize this, we first try to increase $\mathbf{\Lambda}^t$ $\forall t \in (t_0, t_0 + T]$. The following sufficient condition for (4.17) is applied by using (4.15):

$$\mathbf{\Phi}^t \leq \mathbf{\Phi}^{t_0} + \mathbf{\Lambda}^t \leq \mathbf{U}_0^{max} \ \forall t \in (t_0, t_0 + T]. \tag{4.19}$$

This relation leads to the following constraint on $\mathbf{\Lambda}^t$:

$$\mathbf{\Lambda}^t \leq \mathbf{U}_0^{max} - \mathbf{\Phi}^{t_0} \ \forall t \in (t_0, t_0 + T], \tag{4.20}$$

which gives an upper limit for $\mathbf{\Lambda}^t$. To satisfy (4.20) while trying to increase $\mathbf{\Lambda}^t$, $\dot{\mathbf{\Lambda}}^t$ is constrained as follows:

$$\dot{\mathbf{\Lambda}}^t \leq k_t(\mathbf{U}_0^{max} - \mathbf{\Phi}^{t_0} - \mathbf{\Lambda}^t) \ \forall t \in (t_0, t_0 + T], \tag{4.21}$$

with k_t being a positive gain parameter that regulates the rate of the increase of $\mathbf{\Lambda}^t$. Substituting $\dot{\mathbf{\Lambda}}^t$ into (4.21) with its expression in (4.16) yields

$$\sum_{l\in\mathcal{L}} \dot{\alpha}_l^t \mathbf{U}_l^t \leq \beta^t \tag{4.22}$$

with

$$\begin{aligned}\beta^t = k_t(\mathbf{U}_0^{max} - \mathbf{\Phi}^{t_0} - \mathbf{\Lambda}^t) - \mathbf{V}_0^{d,t^T} \nabla_{\mathbf{H}_0^{d,t}} \mathbf{U}_0^t \\ + \mathbf{D}^t - \sum_{l\in\mathcal{L}} \alpha_l^t \mathbf{V}_l^{d,t^T} \nabla_{\mathbf{H}_l^{d,t}} \mathbf{U}_l^t,\end{aligned}$$

for which the following sufficient condition is applied:

$$\dot{\alpha}_l^t \leq \frac{\beta^t}{\sum_{l\in\mathcal{L}} \mathbf{U}_l^t}. \tag{4.23}$$

Relation (4.23) gives an upper limit of $\dot{\alpha}_l^t$.

Finally, based on (4.18) and (4.23), the bounded wrench for each lower-priority task is

$$\mathbf{W}_l = -\alpha_l \nabla_{\mathbf{H}_l} \mathbf{U}_l - \mathbf{B}_l \mathbf{V}_l, \ l \in \mathcal{L}, \tag{4.24}$$

with α_l satisfying:

$$\begin{cases} 0 \leq \alpha_l^{t_0} \leq \min(\dfrac{\mathbf{U}_0^{max} - \mathbf{U}_0^{t_0} - \mathbf{E}^{t_0}}{\sum_{l \in \mathcal{L}} \mathbf{U}_l^{t_0}}, \alpha_l^{max}), & \text{(4.25a)} \\ \dot{\alpha}_l^t \leq \min[\dot{\alpha}_l^{max}, k_\alpha(\alpha_l^{max} - \alpha_l^t), \dfrac{\beta^t}{\sum_{l \in \mathcal{L}} \mathbf{U}_l^t}], & \text{(4.25b)} \\ \text{and } \alpha_l^t \geq 0, \ \forall t \in (t_0, t_0 + T], & \text{(4.25c)} \end{cases}$$

where α_l^{max} and $\dot{\alpha}_l^{max}$ are used to limit the upper limits of α_l and $\dot{\alpha}_l$; and k_α is a gain parameter whose value is user defined. In practice, we choose a value for α_l, which satisfies (4.25a) at the initial time. Then during the simulation, we use (4.25b) to try to increase α_l and, at the same time, to maintain the higher-priority frame inside its admissible domain.

The bounded wrenches will be used as on one hand, the new desired wrenches $\tilde{\mathbf{W}}^d$ in the optimization objectives, and on the other hand, the wrench bounds in the optimization constraints, as is illustrated in Fig.3.3.

4.4 Results

The proposed approach has been implemented on the virtual character described in Section 1.2. The control has been realized in real-time with a simulation time step of $0.01s$.

The CoM task is used for balance control; the hand tasks are used for reaching or manipulation control. The CoM task is chosen as a higher-priority task. The hand task wrenches are bounded to prevent them from driving the CoM out of its admissible domain. The desired CoM position $\mathbf{X}_{com}^d$ is limited by its maximum and minimum values: x^{min}, x^{max}, y^{min}, y^{max}, z^{min}, and z^{max}. There are different ways to define the CoM admissible domain, according to the desired CoM position, its limits, and the hand task targets. Two examples are given in Fig.4.1. In our experiments, the way described in Fig.4.1 (b) is adopted. The reference frame is defined as follows: the x-axis points to the right, the y-axis points to the front, and the z-axis is determined by the right-hand rule. The CoM is inside its admissible domain at the beginning of each simulation.

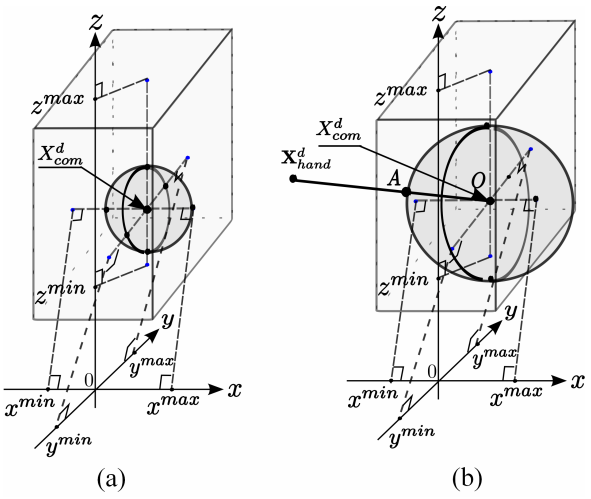


Figure 4.1: Examples of the CoM admissible domain (the shaded ball). The maximum and minimum values of the CoM position form a cube. The boundary of the admissible domain is a sphere, with the origin O at $\mathbf{X}^d_{com}$. (a)Sphere is inscribed in the cube. (b)Radius of the sphere is the length of the line OA starting from $\mathbf{X}^d_{com}$ and pointing toward the hand task target $\mathbf{X}^d_{hand}$. The point A is the intersection point between OA and the cube. If the CoM is very close to x_{max}, then the admissible domain in (a) can be too conservative. In fact, if the hand task draws the CoM toward the direction of x_{min}, then a less conservative admissible domain as in (b) can be used, which is allowed to go out of the cube in some directions.

4.4.1 Reaching

In order to test our approach, the virtual character is first assigned with different reaching tasks. It is required to reach out for some objects with its hands but without moving its feet. The objects are either out of the virtual character's reach (Fig.4.2 left and middle) or on the ground (Fig.4.2 right). The objects that are situated out of the virtual character's reach are used to see if the hand task wrenches can be sufficiently bounded to prevent the virtual character from falling down toward these objects. Our method has been tested with objects on the ground, since the CoM can easily move out of its admissible domain while crouching down.

The results with or without the application of wrench bounds, including the norm of the task force of the right hand and the projection of the CoM position on the ground, are shown in Fig.4.3. These results suggest that our approach can successfully ensure the

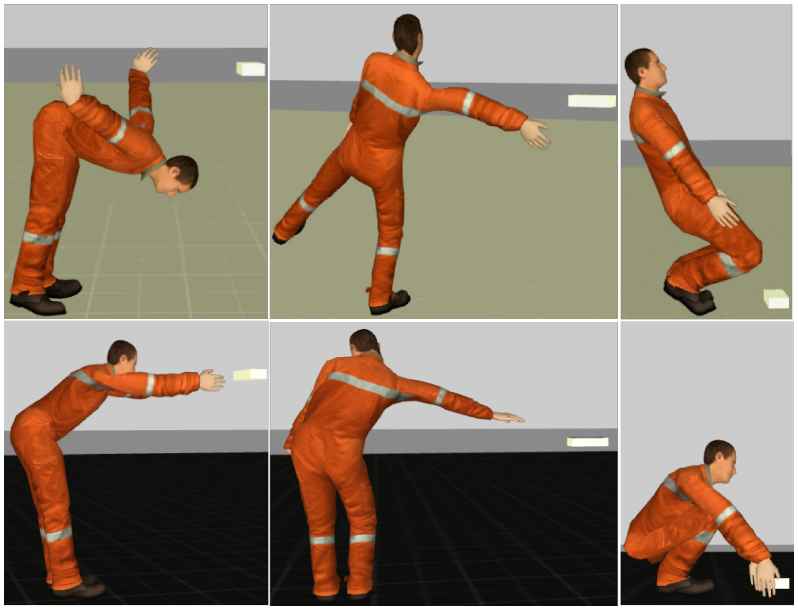


Figure 4.2: Snapshots of the virtual character performing reaching tasks without wrench bounds (above) and with wrench bounds (below). The upper part shows how the tasks are poorly performed without the application of wrench bounds. The virtual character loses its balance for the reaching task to be fulfilled. The lower part shows that when wrench bounds are applied on hand tasks, the virtual character can keep its balance while trying to reach out for the objects.

controller performance of a higher-priority task. Without the constraint of wrench bounds, the norm of the hand task force can be very large. Consequently, the hand task force can drive the virtual character too strongly so that it leans too much toward the objects, and its CoM may move out of the allowed domain. However, with wrench bounds applied on the hand task, the CoM can remain inside the admissible domain throughout the task.

The desired value of each optimization variable has been compared with its value found by the optimization. The results of the task wrench of the right hand during reaching out for an object in front are shown in Fig.4.4. It can be seen that the optimization solution of each variable is very close to its desired value.

4.4.2 Object manipulation through interaction with an operator

We have applied our prioritized control to two object manipulation tasks, all of which are performed by real-time interactions with the operator. The character is required to

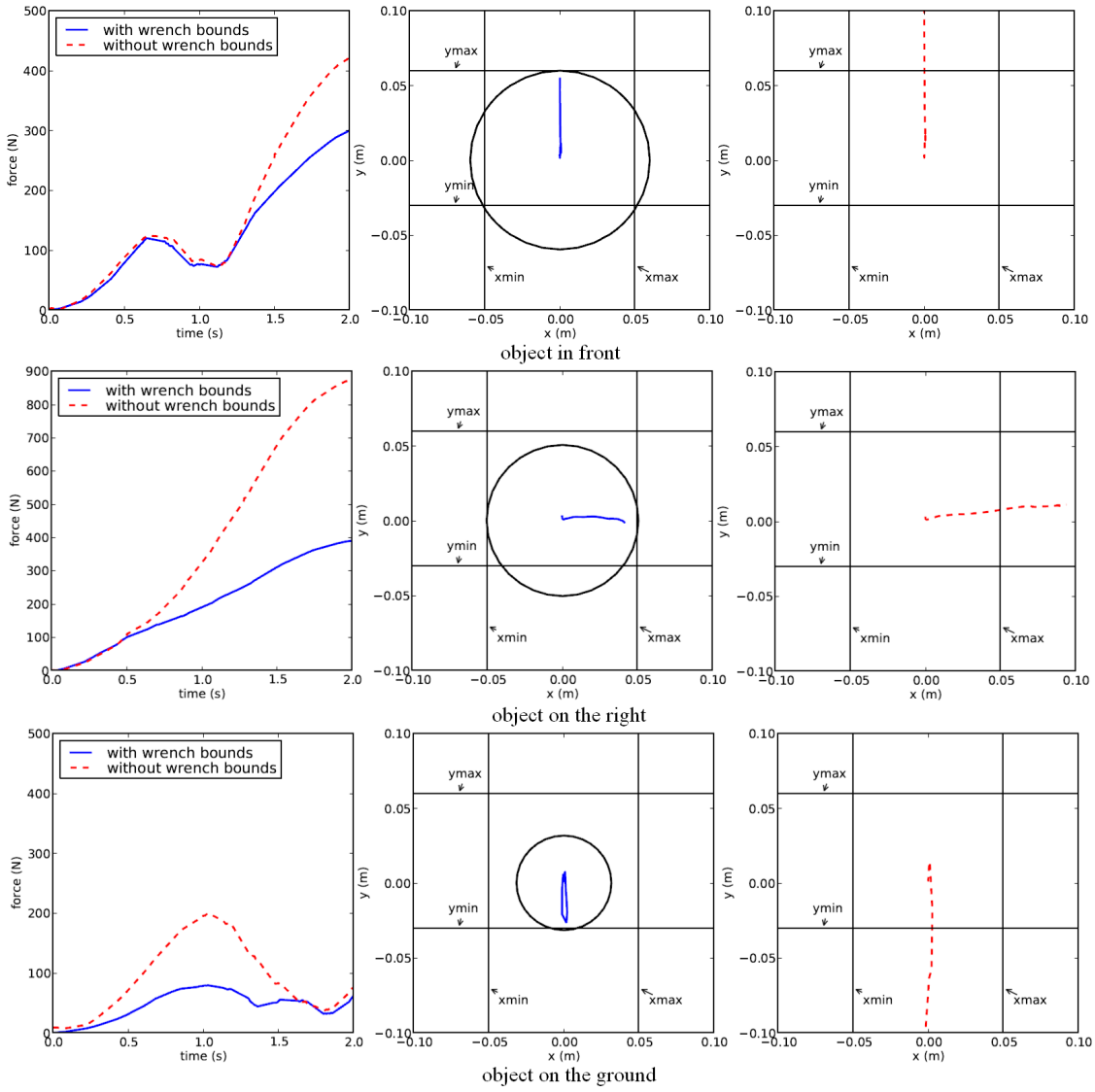


Figure 4.3: Results of reaching for different objects: the norm of the task force of the right hand (left), the CoM position using wrench bounds (middle) and not using wrench bounds (right). The reference trajectory of each hand task is an interpolated trajectory from the initial to the target hand configuration. The admissible domains are shown by the circles.

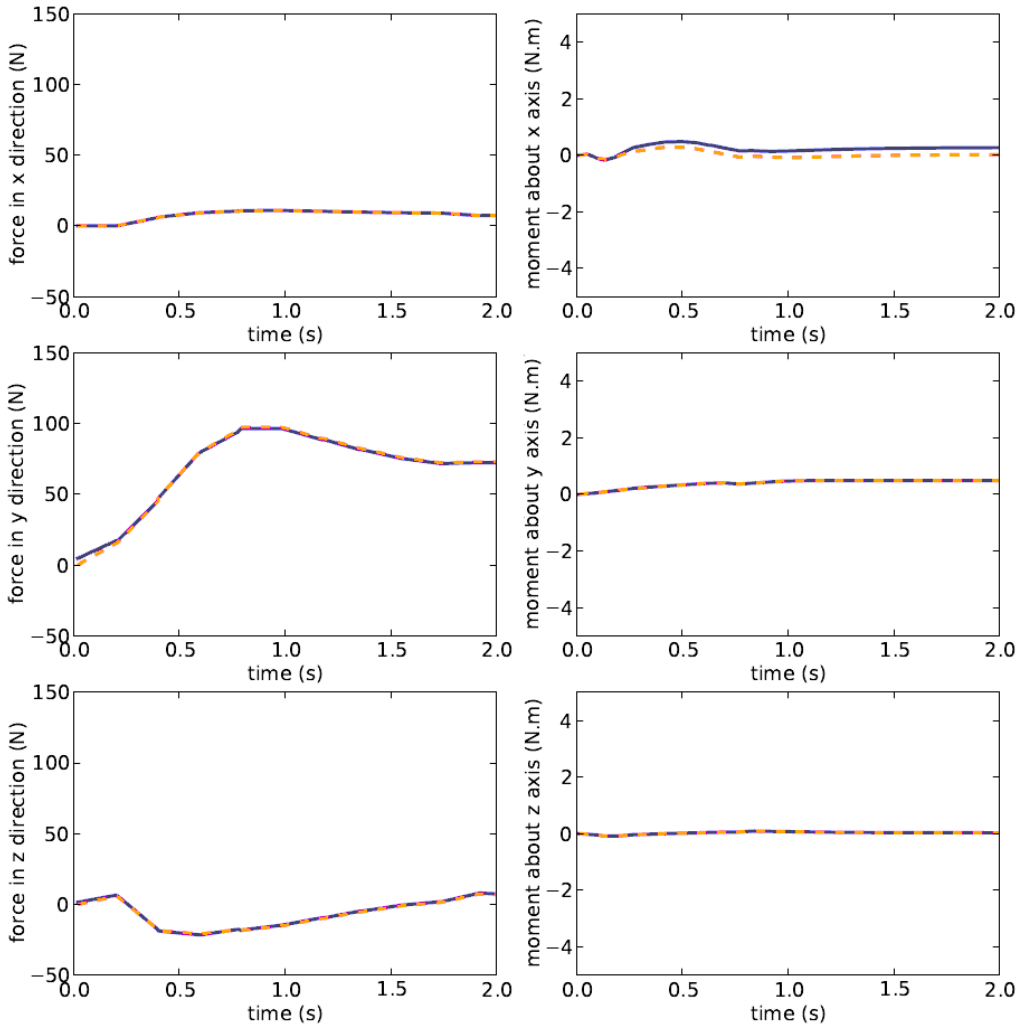


Figure 4.4: The desired hand task wrench $\tilde{\mathbf{W}}^d$ with the application of wrench bounds (dark blue lines) during reaching out for an object in front, and the optimization solution $\hat{\mathbf{W}}$ (orange dashed lines).

crouch down, then turn a lever (Fig.4.5) and a tap (Fig.4.6) located on the ground. Wrench bounds are applied during task execution to ensure balance.

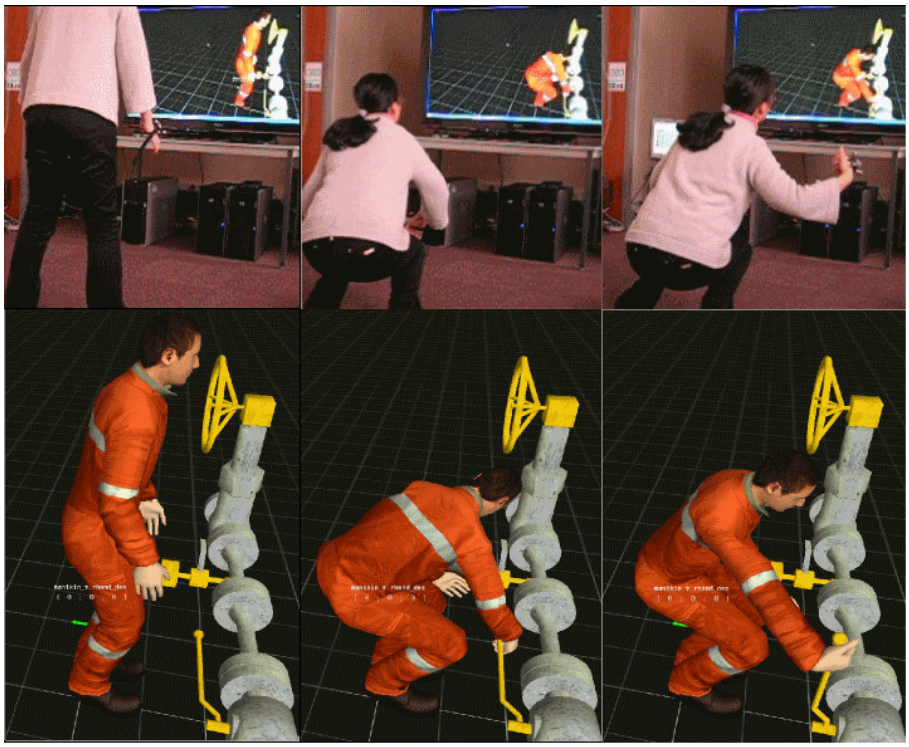


Figure 4.5: Snapshots of the virtual character manipulating a lever.

In the experiments, when the desired values of hand task wrenches are not bounded in the optimization, the character can lose balance from time to time. The reasons for this are, on one hand, a movement of crouching down requires great changes in posture, as a result, the CoM position can often be too close to or even out of its admissible domain, as is shown in Fig.4.3 below; on the other hand, the interaction forces due to object manipulations can perturb the motions of the CoM. However, when bounds are imposed on hand task wrenches, the CoM movements are restricted throughout the manipulation. The virtual character can successfully move its body to manipulate the objects without losing balance.

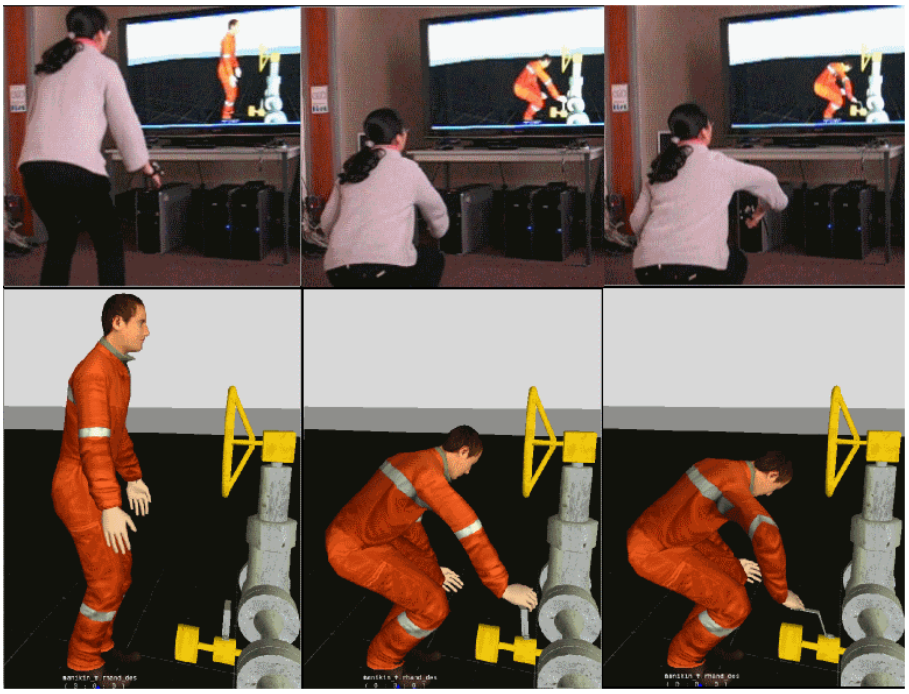


Figure 4.6: Snapshots of the virtual character manipulating a tap.

4.5 Discussion

4.5.1 Verification of the assumption

We have assumed in Section 4.3.3 that for many tasks which do not require very fast motions, the bounded wrenches given by our approach can satisfy the optimization constraints, and contact forces satisfying these constraints can be found. This assumption can be verified by comparing the bounded desired values of the optimization variables with their optimization results, since the former should be equal to the latter if what we have assumed is true. The experimental results, for example those in Fig.4.4, show that the desired values are very close to the optimization results. This suggests that the assumption works well during the experiments.

For motions where the contact constraint (3.3c) can be violated, such as some fast motions, this assumption does not work well, and the passivity of the system may not be guaranteed. Future work should find a way to early predict and avoid such situations.

4.5.2 The choice of control parameters

For many multi-objective control using optimization techniques, the control parameters such as task gains should be chosen very carefully, since they can affect the relative performances between tasks of different priority levels. Even though some gains can work for the tracking of one reference position, they might not be suitable for another one. In other words, it is difficult to manually find task gains that can ensure prioritization satisfactorily for a whole reference trajectory. This problem becomes more obvious during real-time interactions with an operator, where the virtual character is continuously tracking the motions of the operator's hands that can be anywhere in the operator's reachable space, and it is extremely difficult to manually tune these parameters for the whole reachable space. Automated optimization of gains to ensure task priorities has not been realized in many existing approaches yet. Our experiments suggest that our approach based on wrench bounds can handle such problem. In our approach, the control gain $\mathbf{K}_l$ of a lower-priority task is not fixed; on the contrary, it is automatically adjusted on-line by multiplying it with α_l for a better task performance according to the state of the system. An example is shown in Fig.4.7, where the hand task gain is automatically adjusted; and the choice of the optimization weights of tracking task objectives will no longer affect the performance of prioritization.

4.5.3 Comparison with some other approaches

In our approach, the priority of the CoM task over the hand tasks is realized by imposing bounds on hand task wrenches. In fact, the hand task wrenches are bounded by the control of the gains. If the hand task wrench can drive the CoM out of its admissible domain, then the proportional gain of the hand task is reduced to a safe value. It is also possible to handle task priorities by two other different approaches as follows.

- *Approach I*: Constraining the CoM to be within its admissible domain by applying a CoM task force derived from a repulsive potential field. If the CoM is close to the boundary of its admissible domain, then the repulsive force will pull it back.
- *Approach II*: Null-space projection methods.

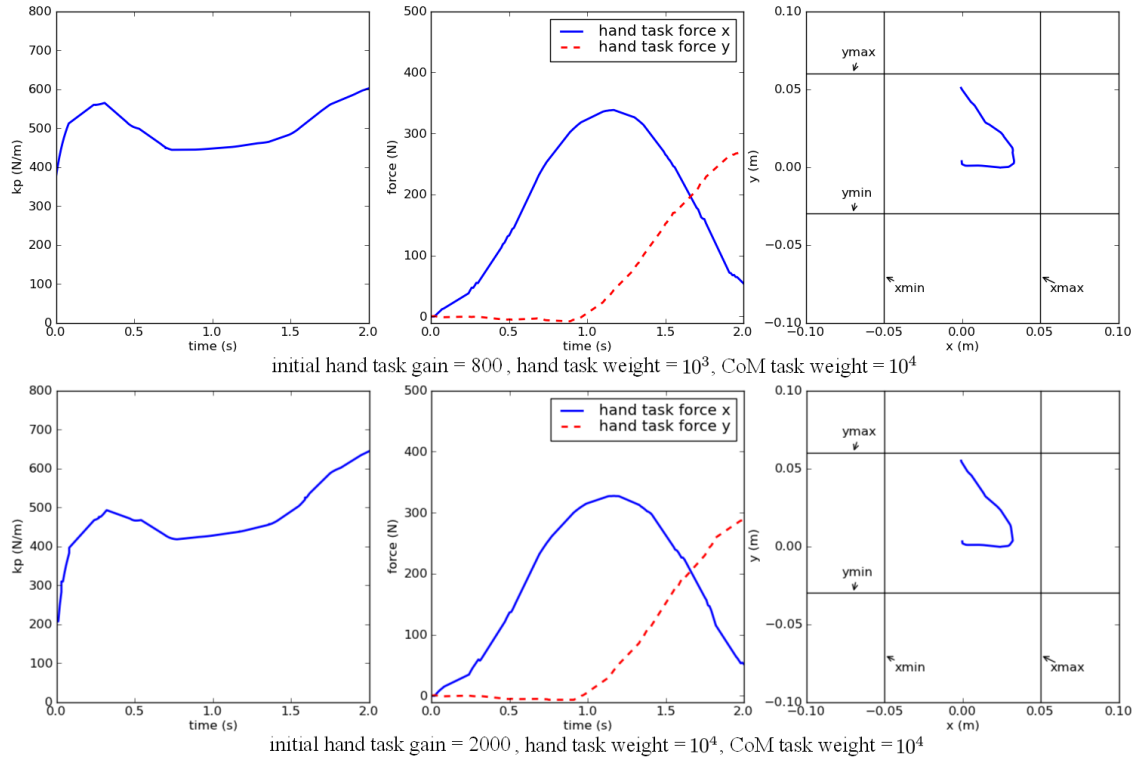


Figure 4.7: Results using different hand task weights and initial gains: the bounded proportional gain $\mathbf{K}_l$ of the hand tasks (left), the bounded hand task force (middle), and the CoM Position (right). Only the elements corresponding to the translational movements of the right hand are shown. The task is to first reach for an object at $1.2m$ on the right with the right hand during $1s$ and then reach for another one at $1.2m$ in front with both hands during $1s$. The bounded task gain varies according to system states during task execution. The CoM remains inside its admissible domain, and the CoM task is allowed to be interfered with hand tasks. The results of prioritization are stable with respect to different choices of initial gains or weights.

Approach I is more simplistic compared with our approach. However, the use of a repulsive potential to constrain the CoM may result in a high CoM task gain. In a simulated environment, the sampling frequency is limited; therefore, a high CoM task gain may lead to unstable motions. This is why our approach decreases the gains of lower-priority tasks, instead of increasing the gain of a higher-priority task.

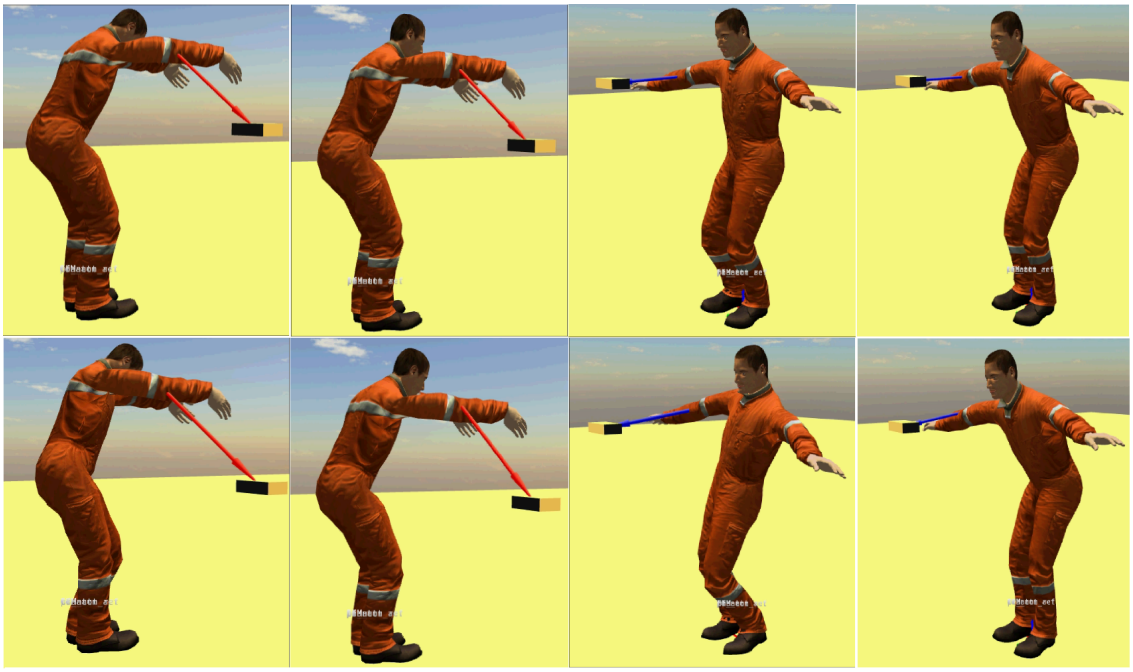
Approach II is a classical method to handle tasks with different priorities. Our approach realizes prioritized control by a quite different way from *Approach II*. The major differences between the two approaches are as follows:

- Our wrench-bound-based control framework provides a quasi-static control; whereas *Approach II* can be dynamically consistent, and it requires the knowledge of the accurate dynamic model of the system.
- Currently, our approach handles task priorities of two levels, while *Approach II* can handle more priority levels.
- As is mentioned before, the passivity of the system is respected by our approach, while projections may cause a risk of breaking the passivity.
- Our prioritized control allows lower-priority tasks to interfere with a higher-priority task in a limited way. This means that instead of being constrained to one desired position, the CoM is allowed to be interfered with a lower-priority task, as long as it remains inside its admissible domain. However, in *Approach II*, lower-priority tasks are not allowed to interfere with higher-priority tasks; as a result, more body movements may be needed to fulfill given tasks.

Our approach has been compared with *Approach II* by some experiments, in which the virtual character is performing reaching tasks. The controller in [Khatib 2008] is used to implement *Approach II*. The same tasks, task priorities, as well as task targets, are used in both approaches. These tasks from the highest priority to the lowest priority are the foot contacts, the CoM, and the hands. The foot contacts are treated as optimization constraints in our approach.

It has been observed that when controlled by our approach or *Approach II*, the virtual character behaves similarly in most cases but differently when mechanical interactions with the environment occurs during task execution. To simulate a mechanical interaction, the virtual character's right arm is connected with an object by an elastic Cartesian string

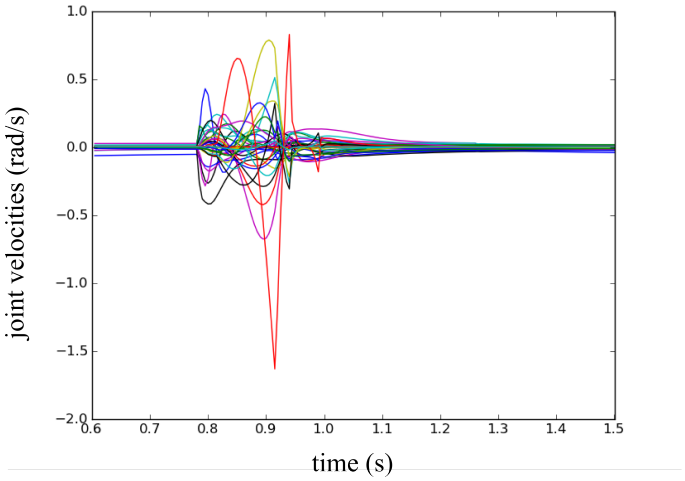
of stiffness $500N.m^{-1}$. A sudden perturbation force is applied on its arm by pulling the object. Note that the perturbation is not strong enough to pull the CoM out of its admissible domain. The resulting behaviors of the character is shown in Fig. 4.8. The



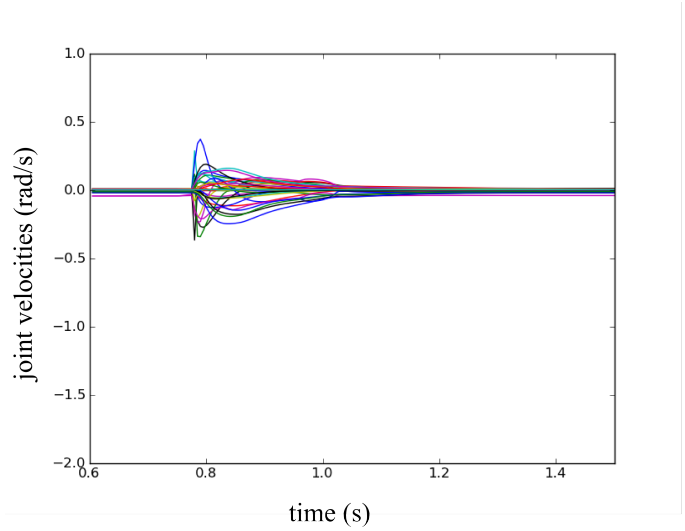
(a) using *Approach II* (b) using our approach (c) using *Approach II* (d) using our approach

Figure 4.8: Behaviors of the virtual character experiencing mechanical interactions with the environment. The virtual character is controlled by *Approach II* in (a) and (c) and by our approach in (b) and (d). The character's right arm is connected with an object by an elastic Cartesian string, which is shown by the arrow. Two tasks are performed: In (a) and (b), the virtual character is moving its hands forward when the object is pulled $0.03m$ rightward and $0.1m$ downward; in (c) and (d), The virtual character is moving its hands toward each side when the object is pulled $0.05m$ forward and $0.05m$ downward. The posture before (above) and after (below) being suddenly pulled by the elastic string are shown. When controlled by *Approach II*, the virtual character rotates its body with its arm pulling the string after perturbation.

reaching behaviors of the virtual character controlled by both approaches are similar before the perturbation. After the perturbation, a lot more body movements are generated with *Approach II*, as can be seen in Fig. 4.8 and Fig.4.9. The whole body rotates and the right arm actively pulls the string. This can be because the hand tasks are not allowed to interfere with the CoM task when the virtual character is trying to compensate for the



(a) Joint velocities using *Approach II*



(b) Joint velocities using our approach

Figure 4.9: Joint velocities of the virtual character controlled by *Approach II* in (a) and by our approach in (b). The character's right arm is connected with an object by an elastic Cartesian string. The virtual character is moving its hands towards each side when the object is suddenly pulled $0.05m$ forward and $0.05m$ downward. After the perturbation, less joint movements are observed when the character is controlled by our approach.

perturbation.

4.6 Conclusions

The proposed approach based on wrench bounds provides a new way to handle two-level task priorities. Such method allows inequality constraints on a higher-priority task; therefore, lower-priority tasks are allowed to interfere with a higher-priority task. Moreover, the passivity of the system has been taken care of to ensure stable operations. As a result, the virtual character can keep the higher-priority task frame inside an admissible domain while trying its best to increase the performance of lower priority tasks in a safe manner. For example, it can keep balance while trying to increase its workspace. The operator can provide any reference motion without being worried about the virtual character's balance problem. The effectiveness of the proposed approach has been demonstrated by experiments on a virtual character performing a wide variety of tasks, such as reaching and object manipulation.

Chapter 5

Putting It All Together

Contents

In this chapter, the posture optimization approach presented in Chapter 2 is combined with the prioritized multi-objective control framework presented in Chapter 3 and Chapter 4. A control system is created, where posture optimization results are adopted by specific task controllers. This control system endows a virtual character with certain levels of autonomy to achieve a long-term goal by performing sequences of tasks. Experiments demonstrate that this control system can autonomously improve task performance.

5.1 Introduction

A main ability of virtual characters that we focus on is to perform human operations in virtual environments, especially in some factory environments, where a virtual character needs to frequently move around and manipulate different objects. The control system presented in this chapter aims at endowing virtual characters with some autonomy, so that they can realize some complex behaviors in such environments.

An autonomous character may possess two levels of autonomy. We refer to the low-level autonomy by the ability to automatically generate appropriate motions given a sequence of tasks to perform. The prioritized multi-objective control framework presented in Chapter 3 and Chapter 4 addresses such kind of autonomy. Although the character is guided by a human operator, our control framework enables it to automatically adjust captured motions to satisfy a variety of constraints in virtual environments. We refer to the high-level autonomy by the ability to automatically select and perform sequences of tasks to achieve a long-term goal. For example, a human operator may send high-level orders to virtual character, such as to move an object from position a to position b. The operator does not provide the character with more information about how to achieve the goal in detail. The character should be able to achieve the long-term goal by autonomously choosing an optimal foot positions for the manipulation task, walking to the optimal position, and finally performing the required manipulation task.

As the problem of realizing low-level autonomy has already been studied in our multi-objective control framework, this chapter will focus on the autonomy of high-level. We investigate a control system, which integrates the posture optimization approach and the multi-objective control framework. This chapter also complements Chapter 2 by demonstrating that our posture optimization approach can improve task performance of a virtual character.

5.2 Control system

The control system used in this chapter is illustrated in Figure 5.1.

Once the human operator gives an order of manipulating an object, the character first defines the manipulation task according to the position and the properties of the object, which are supposed to be known in the virtual environment. The manipulation task is defined by the manipulation path (denoted as $\mathbf{X}_h^d$), the manipulation force direction (denoted as $\hat{\mathbf{F}}_h$), as well as some geometric, kinematic and force related constraints.

Once the manipulation task is defined, the following control is divided into three steps:

– ***Step I*: Off-line posture optimization.**

Before task execution, the posture optimization module is carried out to find suitable

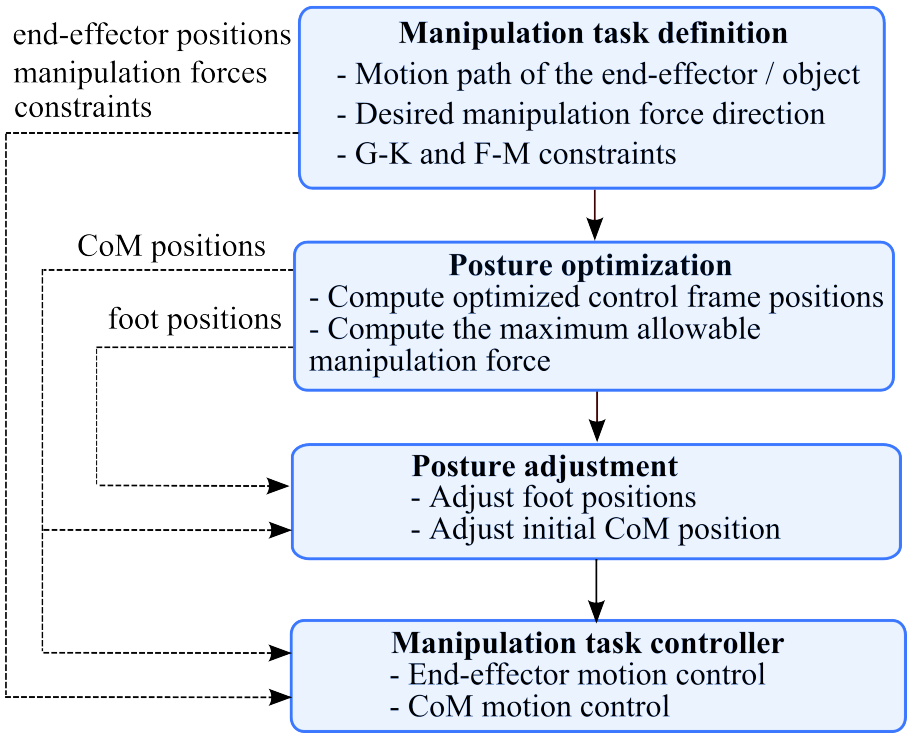


Figure 5.1: Overview of the control system.

postures for the manipulation task. A posture optimization problem is solved based on the actual foot positions of the character and the manipulation task requirements. The optimization result provides us with the optimal solution Θ^*, from which the joint angles (q^*), the positions of control frames, especially the CoM positions ($\mathbf{X}_{com}{}^*$) and the foot positions ($\mathbf{X}_{\mathrm{lf}}{}^*$, $\mathbf{X}_{\mathrm{rf}}{}^*$), as well as the maximum allowable value of the manipulation force magnitude (k^*), can be obtained.

– ***Step II*: On-line posture adjustment.**

Before executing a manipulation task, the posture adjustment module adjusts the contact configuration and the CoM position of the character according to the optimized posture. In our implementation, a walking controller is applied to make the character walk to the optimized position ($\mathbf{X}_{lf}{}^*$, $\mathbf{X}_{rf}{}^*$). The walking motion generator presented in [Herdt 2010] is applied here, which generates automatically the reference trajectories of the CoM and the feet according to their initial states and their desired states ($\mathbf{X}_{com}{}^*$, $\mathbf{X}_{\mathrm{lf}}{}^*$, and $\mathbf{X}_{\mathrm{rf}}{}^*$). These trajectories are used in our multi-objective control framework as the reference trajectories for the tasks of the CoM and the feet to realize the walking motion.

– ***Step III*: On-line manipulation task control.**

The character finally starts to perform the manipulation task by using a manipulation task controller, which takes as inputs the optimized control frame positions and manipulation task requirements. Our multi-objective control framework is applied. The positions of control frames $\mathbf{X}_j(\mathbf{X}^*_{com}, q^*)$ can be used as reference positions in the controller. The joint torques are computed based on the comprehensive consideration of the desired control frame positions, joint angles, contact forces and the gravity force.

A proportional-derivative (PD) control law is applied to compute the desired task force of each control frame j, based on the state error (position error δ_P and velocity error δ_v).

$$\mathbf{F}_j^d = \mathbf{K}_j \delta_{Pj}(\mathbf{X}_j{}^*, \mathbf{X}_j^r) + \mathbf{B}_j \delta_{vj} \tag{5.1}$$

with $\mathbf{K}$ and $\mathbf{B}$ denoting the proportional and derivative gain matrices respectively. The optimized joint angles may also be taken into account by the computation of desired joint torques τ_q^d.

$$\tau_q^d = \mathbf{K}_q \delta_{Pq}(q^*, q^r) + \mathbf{B}_q \delta_{vq} \tag{5.2}$$

Moreover, additional interaction force $\bar{k}\hat{\mathbf{F}}_h$ for manipulating the object is added to the

desired motion task forces of the hand.

$$\mathbf{F}_h^d = \mathbf{K}_h \delta_{Ph}(\mathbf{X}_h^d, \mathbf{X}_h{}^r) + \mathbf{B}_h \delta_{vh} + \bar{k}\hat{\mathbf{F}}_h \tag{5.3}$$

with $k^L \leq \bar{k} \leq k^*$ and k^L denoting the lower bound of the manipulation force magnitude defined by the task.

In the next section, we will show that the manipulation task performance can be improved using the optimized postures.

5.3 Results

The proposed control system has been applied on a virtual character, which should take appropriate actions to achieve some final goals of object manipulation. Each module in our control system plays an important role for the improvement of the task performance of the character. The posture optimization module helps the character to find optimized postures for object manipulation, and to ensure foot positions are suitable for the manipulation task. The prioritized multi-objective control framework helps the character to handle a variety of motion tasks and contact forces. By choosing the CoM task as the higher-priority task, the wrench bounds imposed on end-effector tasks help to maintain the character's CoM inside a safe domain to ensure its balance.

In Chapter 2, our posture optimization approach has been tested on a simplified model of the virtual character; while in this section, this approach is implemented on our virtual character. Since the effects of our prioritized multi-objective control framework on the virtual character's task performance have already been discussed in previous chapters, the experimental results here will be presented with respect to the effects of our posture optimization approach.

To show how the control system with a preparation stage using the posture optimization approach can help to improve task performance, each of the following tests are divided into two parts:

- In the first part, the virtual character tries to perform the task without using posture optimization results, with its feet remaining at their initial positions.
- In the second part, the control system presented in Section 5.2 is applied. The char-

acter first searches for optimal postures, then adjusts the foot positions according to their optimal values, and finally performs the manipulation tasks according to the optimal control frame positions. The experiment setup for each manipulation task is in Table 5.1.

Table 5.1: Experiment setup: desired motion path $\mathbf{X}_h$ and manipulation force direction $\hat{\mathbf{F}}_h$ (applied by the objects on the character) associated with each discretized point, where $\mathbf{O}_{\text{obj}}$ denotes the origin of the valve, and $\mathbf{X}^0_{\text{obj}}$ denotes the initial contact position between the hand and the object.

Task	$\mathbf{X}^i_h$, $\hat{\mathbf{F}}^i_h$
open a valve	$\mathbf{O}_{\text{obj}}+[0,0,0.25]^T$, $[-1,0,0]^T$
	$\mathbf{O}_{\text{obj}}+[0.177,0,0.177]^T$, $[-0.707,0,0.707]^T$
	$\mathbf{O}_{\text{obj}}+[0.25,0,0]^T$, $[0,0,1]^T$
move box	$\mathbf{X}^0_{\text{obj}}$, $[0,0,-1]^T$
	$\mathbf{X}^0_{\text{obj}}+[0,-0.2,0.6]^T$, $[0,0,-1]^T$
	$\mathbf{X}^0_{\text{obj}}+[0,0.2,0.6]^T$, $[0,0,-1]^T$
push storage cabinets	$\mathbf{X}^0_{\text{obj}}$, $[0,-1,0]^T$
	$\mathbf{X}^0_{\text{obj}}+[0,0.4,0]^T$, $[0,-1,0]^T$

5.3.1 Follow a desired motion path

Our approach can choose suitable postures that allow the character's hand to follow a curved motion path. To demonstrate this, the character has been required to open a valve with a radius of $0.25m$ to 90 degrees with his right hand. The hand should follow exactly the given motion path as quarter of a circle, because the valve can only rotate around its rotation axle which is fixed. The desired hand force is tangential to the motion path. The initial foot positions are not optimized for the task, as they can cause a break of foot contacts during manipulation. The behaviors of the character using non optimized postures and optimized postures are shown in Figure 5.2 and Figure 5.3 respectively.

When using non optimized postures, the foot positions can not support the whole manipulation path. The contacts between the feet and the ground are broken. The optimized

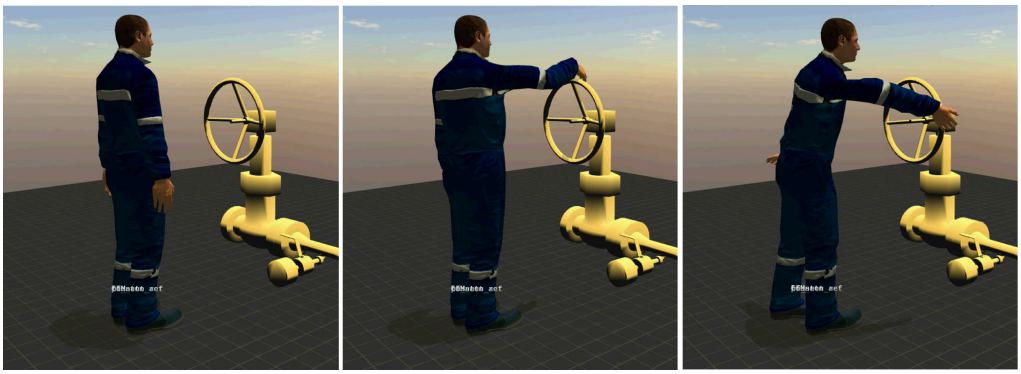


Figure 5.2: The virtual character opening a valve using non-optimized postures.

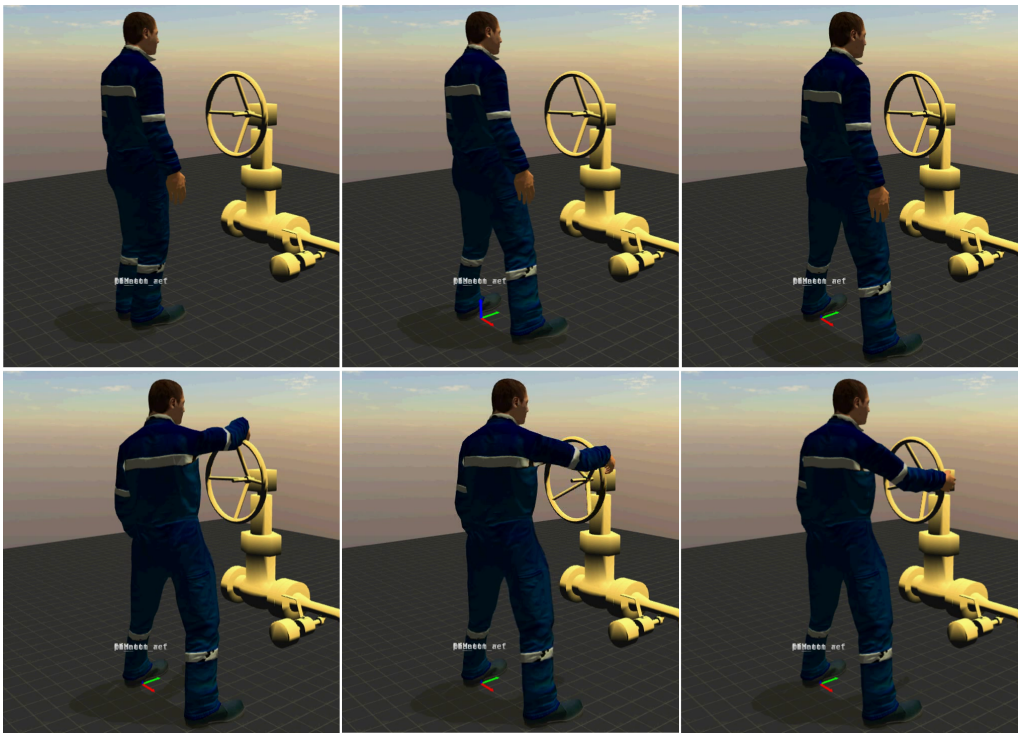


Figure 5.3: The virtual character opening a valve using optimized postures.

foot positions make the character walk rightward before starting manipulation. This foot contact configuration enables the hand to open the valve along the given motion path without breaking foot contacts. We have also observed that less upper body movement is generated by using the optimal postures than not using them, which makes the whole body motion more natural.

5.3.2 Obstacle avoidance

An experiment of moving an object while avoiding obstacles has been conducted. The character should change postures so as to allow the hands to easily approach a box located between a table and a shelf above the table, then move it around the shelf, and finally put it on the shelf (Figure 5.4). The desired manipulation path goes around the shelf with a

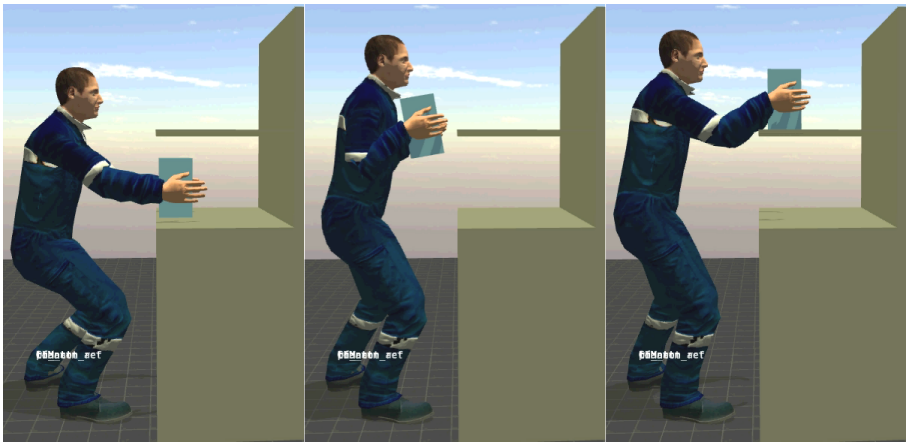


Figure 5.4: The virtual character moving a box while avoiding obstacles.

safety margin. The postures optimized according to this manipulation path can allow the character to successfully fulfill the task without causing collisions with the shelf.

5.3.3 Joint comfort

Our posture optimization provides posture solutions that can take into account the joint discomfort measure. The function of the joint discomfort objective is to enhance the preference of using certain joints to fulfill a motion task. This function can be observed by comparing the behaviors of the character taking up light and heavy box. In the experiments, the neutral values q^N are set to joint angles of an erect standing posture.

(a) $m = 5kg, \phi_{back} = 0.4$ **(b)** $m = 15kg, \phi_{back} = 10$

Figure 5.5: The virtual character lifting box of different mass (m).

A higher value of a joint weight ϕ_l reinforces the value of q_l to be closer to q_l^N, which means we prefer to use joints with weights lower than ϕ_l to make the end-effector attain the desired position. Similar to [Yang 2004], we set higher weights for joints on the back of the character, and lower weights for joints on the arms and the legs. In addition, we adjust certain joint weights to adapt to different task requirements. For a task of lifting a box, the character can either lean over with the back then lift it up using the back, or crouch down while keeping the back straight then lift it by standing up. People tend to choose the latter one to take up a heavy object. This is what physical therapists usually suggest people to do in order to protect their backs. Such behaviors can be achieved by tuning the value of joint weights. For heavy box, high values are assigned to ϕ_{back} associated with joints on the back of the body. As a result, the character just slightly crouches down to pick up a light box; whereas it crouches down more and carefully keeps the back straight to take up a heavy box (Figure 5.5).

5.3.4 Handle interaction forces

When searching for suitable postures for the task of lifting a box, the weight of the box can be taken into account by k^L in (2.15), which indicates the minimum force that is necessary to lift the box. We have observed that when the box is heavy, the optimized CoM position is obviously behind its initial position. This result is consistent with the needs of the character to lean backward so as to balance the interaction force due to the

weight of the box.

Similar results have been observed during pushing tasks. The character has been required to push forward a storage cabinet (Figure 5.6) a distance of up to $0.4m$. Storage cabinets of different mass m (from $30kg$ to $50kg$) and different friction coefficients with the ground μ (from 0.1 to 0.4) are used.

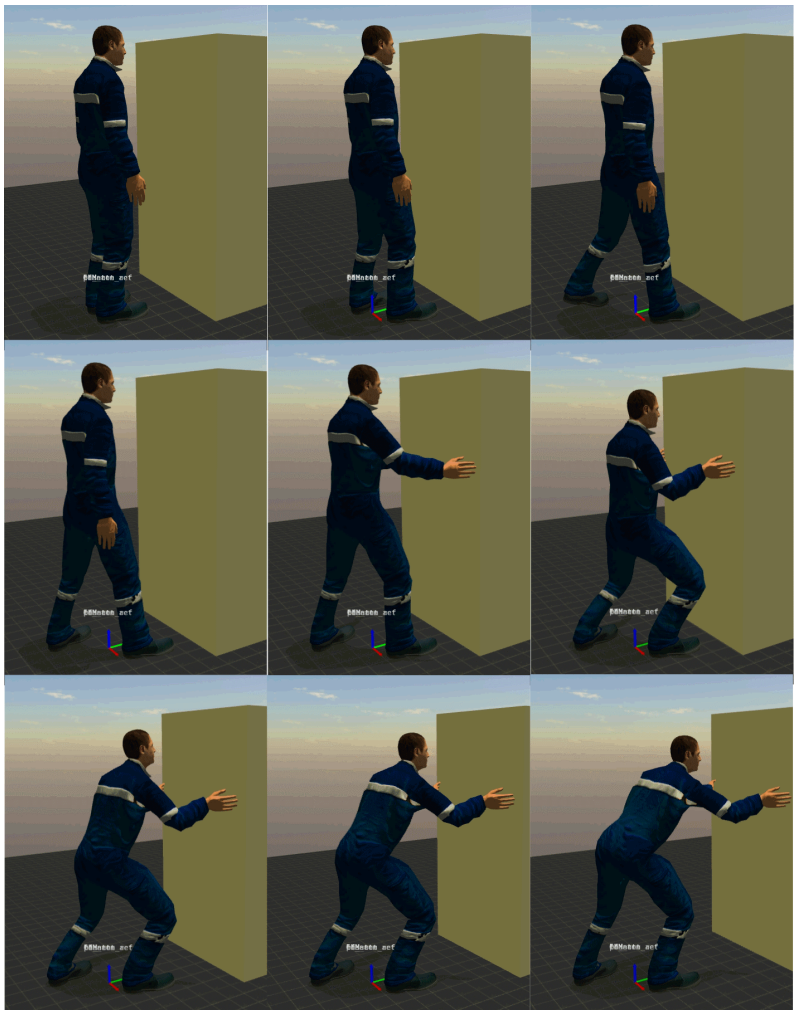


Figure 5.6: The virtual character pushing a storage cabinet using optimized postures.

The results of optimal foot positions tell the character to separate the feet along the pushing direction, so as to generate a robust posture against the pushing force. Moreover, the optimal CoM and shoulder positions tell the character to lean forward. Similar behaviors can be observed when a real human attempts to push strongly.

More fluctuation of the interaction force has been observed without the use of optimal

postures, which suggests that optimal postures help to generate more coherent motion during manipulation. The magnitudes of the forces applied by the right hand on one storage cabinet during pushing tasks are shown in Figure 5.7. It can be seen that the

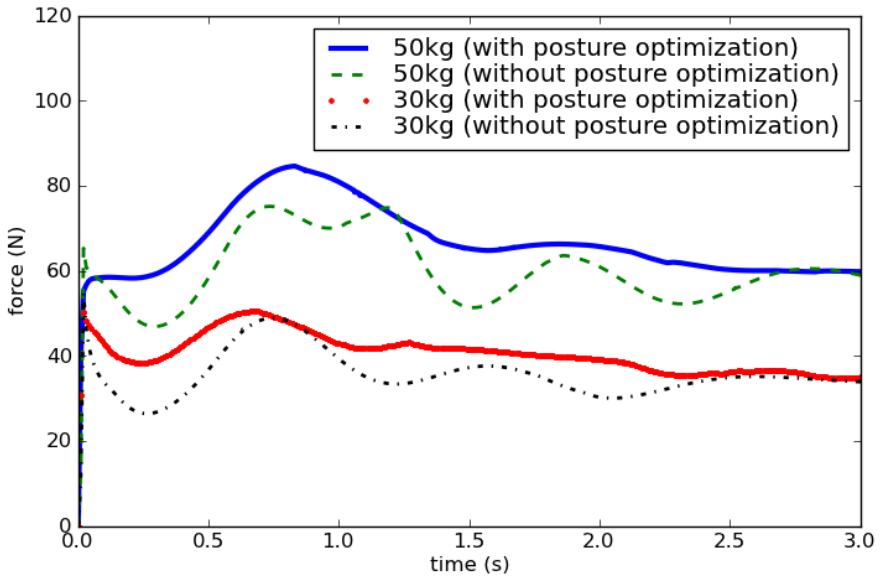


Figure 5.7: Results of the force applied by the hand on the storage cabinet.

interaction forces resulting from optimal postures is more stable than those resulting from non-optimal postures. Without an optimization before the task execution, the character may find his posture not quite adaptable for the task from time to time. If continuing pushing forward as strongly as before will result in the loss of balance, then the character will sacrifice the hand task performance to ensure its balance; because the balance task is of higher priority than all the other motion tasks. Consequently, the pushing force will be reduced at this moment. However, task performance can be improved by using the optimized CoM positions as the reference positions in the manipulation task controller, because the optimized CoM positions provided by our posture optimization are suitable for the contact configurations and the interaction forces.

5.3.5 Robustness to mechanical interactions

Generally, the optimal postures tell the character to increase the distance between its feet, along the direction of the manipulation force, so as to generate a robust pos-

ture against this force. For example, when opening a valve, its feet are separated from each other mainly in the lateral direction. However, when moving an object forward or backward, the feet are much more separated in the sagittal direction.

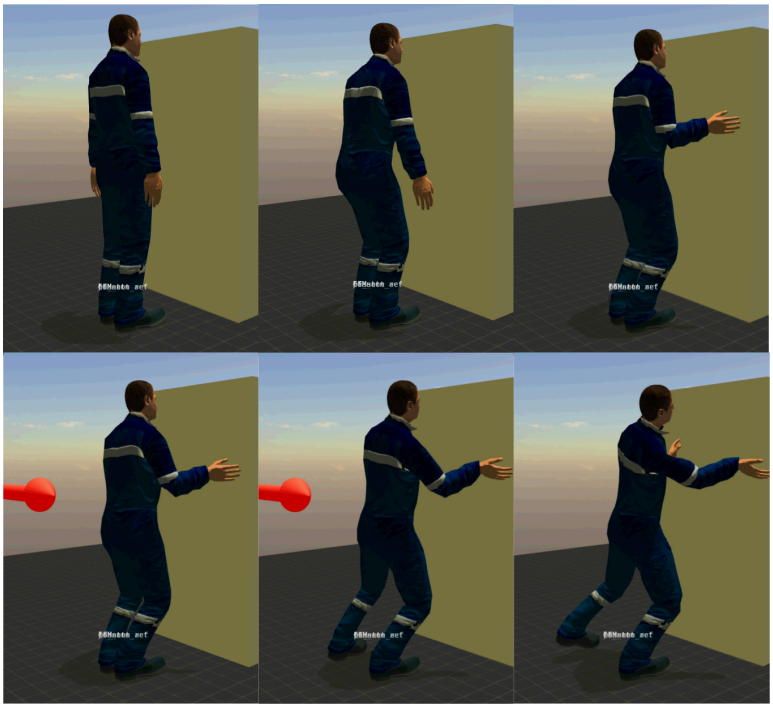
As mentioned in 2.3.3, perturbations can be taken into account in posture optimization. In our experiment, some perturbation forces, including those which are perpendicular to the manipulation force, are considered. During task execution, external perturbation forces (up to $120N$ during $0.5s$ or up to $90N$ during $1s$) have been applied on the character (Figure 5.8). When pushed by strong perturbation forces, the character using non-optimal postures may abandon the task and sometimes lose its balance. It has to move its foot to try to keep its balance. However, by adopting optimal postures, the character's ability to continue task execution under some perturbation forces can be enhanced. Less body movements have been generated to resist strong pushes and to recover from them, and neither foot slipping nor a break of foot contacts has been observed.

5.3.6 Perform tasks through interactions with a human operator

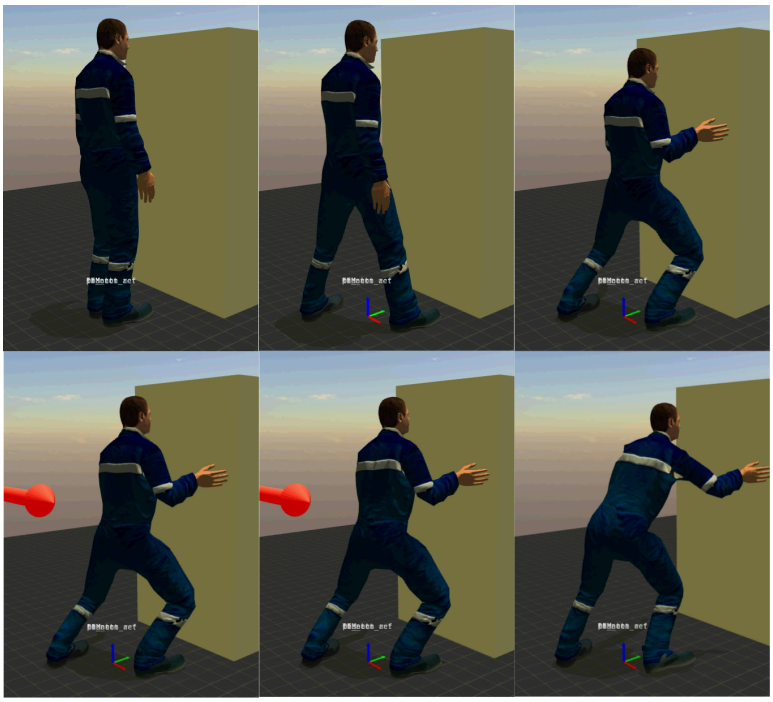
In the above-mentioned experiments, posture optimization problems have been solved according to some desired manipulation trajectories, which are preprogrammed for the manipulation tasks. In fact, the proposed control system can also make the character execute an operator's orders through real-time interactions, and at the same time, try to improve the task performance. In the implementation presented in [Nguyen 2010], the operator needs to adjust his postures according to the character's balance features, such as the support polygon and the CoM. Our approach automatically computes adaptable postures of the character in advance for a task, and then further adjusts its postures during task execution, so that the operator does not need to compensate for the character's balance. To demonstrate this, the character has been required to fulfill some final goals such as turning a value, moving a box, and pushing a storage cabinet, in an interactive manner. The resulting behaviors of the virtual character during these experiments are shown in Fig.5.9, Fig.5.10, and Fig.5.11.

The hybrid control coupling the control of the operator and the control of the proposed robotic control system is realized in the following way.

– At the beginning of each experiment, the posture optimization approach is applied



(a) Before, during, and after perturbations using non-optimized postures



(b) Before, during, and after perturbations using optimized postures

Figure 5.8: Behaviors of the character suffering from an external perturbation force (shown by the red arrow) of $120N$ during $0.5s$ when pushing a storage cabinet. The optimal postures can better resist the perturbation.

Figure 5.9: The virtual character opening a valve through interactions with an operator.

Figure 5.10: The virtual character moving a box through interactions with an operator.

Figure 5.11: The virtual character pushing a storage cabinet through interactions with an operator.

to find optimized postures according to the object to manipulate. In our current implementation, each object to manipulate is related to one kind of manipulation task; and a reference manipulation path is given a priori. For example, the character knows that it can only "turn" the valve, "take up and move" the box, "push" the storage cabinet. It will not think about "pushing" the valve or "taking up" the storage cabinet.

- When optimized foot positions have been found, the character walks to the desired positions.
- Once arrived at the desired position, the character starts to observe the motions of the operator. For interactions with the operator during object manipulation, a finite state machine (FSM) like the one presented in Fig.3.7 in Section 3.5 is implemented. During the *Tightening* state, desired contact forces are added to the desired hand task wrenches to enhance the contacts with an object. During each state, the CoM task target takes the optimized value found during the posture optimization stage, instead of being manually chosen a priori. For example, during *Manipulating* state, the CoM automatically moves slightly backward to balance the weight of the box; or automatically moves forward to generate a stronger pushing force against the storage cabinet.

In our current implementation, for each FSM state (except for the idle state), the task controller uses the reference motions, which are the smooth interpolations of the optimized postures computed by posture optimization. The character tries to follow the reference motions with predefined reference velocities. The operator's motions are used to trigger the state transitions in the FSM. Another way to realize an interactive control is to make a projection between the operator's motions and the optimized reference motions, and control the character to move along the reference motion path while trying to follow the references that is associated with the operator's movements. In this way, the operator's motions can have two functions: on one hand, they can be used to trigger state changes; on the other hand, they can guide the motion velocities of the character.

The experiments suggest that, by using optimized postures, the character can accomplish the required task and improve task performance, while avoiding failures due to poor postures of the operator.

5.4 Conclusion

The experiments presented in this chapter suggest that the control system, which combines the posture optimization approach with the prioritized multi-objective control framework, allows the virtual character to autonomously achieve a long-term goal of object manipulation. The virtual character can autonomously choose an optimal foot positions for the manipulation task, walk to the optimal position, and finally fulfill the required manipulation task.

The experiments suggest that this control system can improve the behaviors of the character in several ways. First, the character can autonomously decide its foot positions with respect to the object to manipulate. The human operator does not need to manually choose such contact configurations for the character. Second, task performance can be improved by choosing suitable postures in the preparation stage before actually performing a task. The optimized postures can ensure that constraints such as joint limits, non-sliding contacts, and geometrical relations with the environment can be satisfied. The use of optimized postures in our multi-objective control can enable the end-effector to follow given manipulation path while applying the maximum manipulation forces without causing foot slipping and balance problems. Moreover, the robustness of postures can be improved, so that character can cope with perturbations due to mechanical interactions.

Chapter 6

Conclusion and Future Work

Contents

6.1 Summary

In this dissertation, we have developed a hybrid control system for interactive virtual characters. This work builds on the research in both motion control approaches in the domain of robotics and motion synthesis approaches in the domain of computer graphics.

Our virtual characters are designed to be interactive with human operators; and they should act in virtual environments where a lot of mechanical interactions are to be handled. Since human operators do not experience the same mechanical interactions in the real world as the characters do in their virtual world, it is not desirable to make a character follow an operator's motions by simply restoring the captured motions of the operator. Based on these facts, we believe that virtual character with certain levels of autonomy can better handle the interactions with both human operators and virtual environments.

We developed a posture optimization approach in Chapter 2, which can be applied in the preparation for manipulation tasks. Our contribution here is on the formulation of

an optimization problem to find optimal and robust postures, including contact positions, with respect to manipulation tasks. This optimization problem takes into account not only geometric and kinematic constraints, but also force and moment constraints. The comprehensive consideration of these constraints allows the virtual character to apply manipulation forces as strongly as possible, to avoid foot slipping, to improve body comfort, and to respect its joint limits and geometrical relations with the environment.

We developed a control framework in Chapter 3, which can drive virtual characters to accomplish a variety of complex behaviors in a physics-based environment. Our contribution here is on proposing a constrained optimization problem, which can be solved on-line to efficiently compute optimal task wrenches and joint torques for finding a compromised solution between following an operator's motions and handling interactions with virtual environments. We have shown that multiple task objectives, multiple constraints, and the redundancy of postures can be handled by solving this optimization problem. One advantage of this control framework based on the Jacobian-transpose method over some acceleration-based control approaches is that the numerical singularities due to the computation of the inverse of the Jacobian can be avoided. We have demonstrated that, by taking captured motions as inputs, this control framework can make a character interact with a human operator in real-time. The character can try to follow captured motions while autonomously assuring the physical consistency of its motions.

We developed in Chapter 4 a novel two-level prioritized control with wrench bounds, which provides a new way to handle multiple task goals of two priority levels simultaneously. Our main contribution on this subject is on proposing a method to compute bounds of lower-priority task wrenches, so as to ensure the controller performance of higher-priority tasks. In contrast with some other approaches, the energy of the system is bounded in our prioritized control, which ensures the passivity of the system, so as to guarantee stable operations. Moreover, inequality constraints on a higher-priority task are supported, which means the higher-priority task frame is allowed to move as long as the error remains within a tolerance margin. This feature makes the constraint on the higher-priority task less restrictive, so that lower-priority tasks are provided with more freedom of movement, as they are to some extent allowed to dynamically interfere with the higher-priority task. By choosing the CoM task as the higher-priority task, the char-

acter can keep balance while trying to increase its workspace. As a result, an operator can interact with the character without the necessity of compromising the character's balance.

Based on the approaches presented in Chapter 2, Chapter 3, and Chapter 4, we have shown in Chapter 5 how all these approaches can be integrated in one control system to endow virtual characters with the capability of automatically performing sequences of tasks to achieve a long-term goal. The combination of all these approaches can greatly expand the autonomy of virtual characters, allowing human operators to control them from a high level by just specifying the long-term goal, without the necessity of providing them with detailed information, such as the motion tasks to perform at each time step and the suitable postures for performing these motions. We have successfully used this control system to improve task performance of a virtual character which moves around and manipulates different objects.

6.2 Future work

6.2.1 Towards more complex behaviors

One possible extension of this dissertation work is to realize more complex behaviors, such as manipulation during walking. These complex behaviors need motion planning, for which the predictive control [Muico 2009,Da Silva 2008a] may be adopted in our framework.

6.2.2 Towards a multi-level prioritized control

Our present system can still be improved by extending our two-level prioritized control with wrench bounds to a prioritized control that can handle multiple priority levels. It is possible to achieve this by the computation of a sequence of the maximum allowed potential energy associated with different priority levels.

6.2.3 Improving the task performance during manipulation

One important future direction is to improve the task performance of the character, by developing approaches to automatically adjust control parameters such as gains and

optimization weights. Another direction is to endow the character with the ability to improve its task performance by anticipation, according to the detected intentions of the operator during interactions.

6.2.4 Realizing object manipulation in a generalized way

To perform an object manipulation task, the character should make a sequence of decisions to handle the following problems:

(a) where to put its feet in order to reach and manipulate the object,

(b) along which trajectory the hands should approach the object,

(c) which grasp type to use to hold the object,

(d) where to hold the object,

(e) how much force to apply on the object, and

(f) how much joint torques to apply during the manipulation.

In our current implementation, the problem (a) is addressed by our posture optimization approach; the problem (f) is addressed by our multi-objective controller; the decisions on problems (b) and (d) can either be made according to human operator's motions or be preprogrammed; and the decisions on problems (c) and (e) are preprogrammed. For example, if we want a virtual character to push a storage cabinet, then we manually choose the pushing controller for the character and tell it how much pushing force to apply; if we want the character to turn a valve, then we manually choose the turning controller for the character and tell it how much turning force to apply.

As objects with different shapes, sizes, and poses may require different grasp strategies, a future research direction is to enable our control system to realize object manipulation in a generalized way by implementing some **automatic grasp generators**. One possible way to allow the character to automatically decide which grasp type to use is to construct an information database that associates the characteristics of the objects with grasp types, such as in [Miller 2003], where objects are simplified into a set of shape primitives, and the grasp strategies are determined using some rules defined for the shape primitives.

It is also possible to implement **smart objects** [Kallmann 1998] that allow the character to choose suitable operations, or to implement **robot programming by demon-**

stration (PbD) [Guenter 2007,Ekvall 2007,Calinon 2009,Stulp 2010,Theodorou 2010, Pastor 2011,Gribovskaya 2011] to enrich our elementary tasks.

6.2.5 Developing a generic method for intention detection

For a virtual character interacting with a human operator, the character's actions are determined based on events from the operator and the simulation environment. The character needs to decide what action to perform by analyzing the motions of an operator. Currently, our virtual character decides how to react according to a finite state machine, which fixes the actions of the virtual character for a given task. Our current implementation can be improved by the development of a more generic method for intention detection. This means that instead of behaving according to manually fixed actions, the character should be able to extract the features of the operator's motions, and based on which, to understand what the operator wants it to do. For example, during the whole time of the interaction with an operator, it would be interesting if the character can continuously detecting whether the operator wants it to take up an object, to push the object, or to release the object. We can make the character learn to detect the operator's intentions by using some learning techniques based on some data mining and clustering approaches, such as Hidden Markov Models [Rabiner 1989,Rigoll 1998], Gaussian Mixture Regression [Sung 2004], Principal Component Analysis [Birk 1997], Dynamic Time Warping [Müller 2005,G. A. ten Holt 2007], and so on.

6.2.6 Towards more complex task transitions

There are two directions for the improvement of our current task transition. The first direction consists in the improvement of the transition graph for the determination of more complex task transitions. As mentioned in Chapter 5, our virtual character may sometimes need to autonomously perform sequences of tasks to achieve a long-term goal required by the operator. Currently, we use a finite-state machine for the transitions among several basic controllers, such as the walking controller and the manipulation task controller. To improve our current task transition, a dynamic transition graph such as the one used in the controller composition system presented in [Faloutsos 2001], or a decision making engine

based on the use of fuzzy logic [Salini 2012], can be constructed for the determination of more complex task transitions. The second direction consists in the regulation of task controllers to avoid abrupt changes in joint torques or discontinuities in motions. This research can be related to the work presented in [Lee 2011,Salini 2011].

Appendix A

Mathematical Proofs

A.1 Rotational potential energy

Here, we prove the non-negative property of the rotational potential energy $\mathbf{U}_l^{\text{rot}}$ defined in (4.3). The rotational stiffness matrix and the associated co-stiffness matrix are denoted as $\mathbf{K}_l^{\text{rot}} = diag(k_1, k_2, k_3)$ and $\mathbf{G}_l^{\text{rot}} = diag(g_1, g_2, g_3)$, respectively. According to (4.4), we have

$$\begin{aligned} g_1 &= (k_2 + k_3 - k_1)/2, \\ g_2 &= (k_3 + k_1 - k_2)/2, \\ g_3 &= (k_1 + k_2 - k_3)/2. \end{aligned} \tag{A.1}$$

Without loss of generality, let $\mathbf{R} = \mathbf{R}_l^{-1}\mathbf{R}_l^d$ denote a matrix for a rotation by an angle of θ, about an axis in the direction of $u = [u_x u_y u_z]^T$, which is written as follows:

$$\mathbf{R} = \begin{bmatrix} c_\theta + u_x^2(1-c_\theta) & u_x u_y(1-c_\theta) - u_z s_\theta & u_x u_z(1-c_\theta) + u_y s_\theta \\ u_y u_x(1-c_\theta) + u_z s_\theta & c_\theta + u_y^2(1-c_\theta) & u_y u_z(1-c_\theta) - u_x s_\theta \\ u_z u_x(1-c_\theta) - u_y s_\theta & u_z u_y(1-c_\theta) + u_x s_\theta & c_\theta + u_z^2(1-c_\theta) \end{bmatrix} \tag{A.2}$$

with $c_\theta = cos\theta$ and $s_\theta = sin\theta$. With these notations, the rotational potential energy can be written as

$$\begin{aligned} \mathbf{U}_l^{\text{rot}} &= tr(\mathbf{G}_l^{\text{rot}}) - tr(\mathbf{G}_l^{\text{rot}}\mathbf{R}) \\ &= (g_1 + g_2 + g_3) \\ &- (g_1(c_\theta + u_x^2(1-c_\theta)) + g_2(c_\theta + u_y^2(1-c_\theta)) + g_3(c_\theta + u_z^2(1-c_\theta))) \\ &= (g_1 + g_2 + g_3 - g_1u_x^2 - g_2u_y^2 - g_3u_z^2)(1-c_\theta) \end{aligned} \tag{A.3}$$

Since u is a unit vector, we have $u_x^2 + u_y^2 + u_z^2 = 1$. Applying this in (A.3) leads to

$$\mathbf{U}_l^{\text{rot}} = (g_1(u_y^2 + u_z^2) + g_2(u_z^2 + u_x^2) + g_3(u_x^2 + u_y^2))(1-c_\theta) \tag{A.4}$$

Then applying (A.1) in (A.4) leads to

$$\mathbf{U}_l^{\text{rot}} = (k_3 u_z^2 + k_1 u_x^2 + k_2 u_y^2)(1 - c_\theta) \tag{A.5}$$

All the terms in the right member in (A.5) are non-negative; therefore, $\mathbf{U}_l^{\text{rot}}$ is non-negative.

A.2 Objective measure to assure slowness

To provide an idea of objective measure to assure slowness, an analysis on the relation between motion slowness and contact constraints is given here. The Linear Inverted Pendulum Plus Flywheel Model (LIPPFM) proposed in [Pratt 2006] is adopted. This model abstracts a biped system as an inverted pendulum with an inertial flywheel centered at the CoM (Fig.A.1), which has a constant height of z_0. The equations of motion are

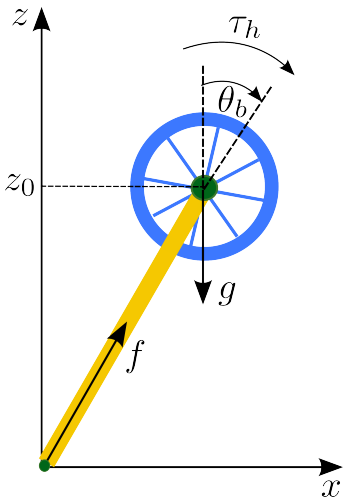


Figure A.1: The Linear Inverted Pendulum Plus Flywheel Model

$$\begin{aligned} \ddot{x} &= \frac{g}{z_0} x - \frac{1}{m z_0} \tau_h \\ \ddot{\theta}_b &= \frac{1}{J} \tau_h \end{aligned} \tag{A.6}$$

where x and z are the CoM horizontal and vertical coordinates; θ_b is the flywheel angles with respect to vertical; g is the gravitational acceleration constant; m and J are the mass and the rotational inertia of the flywheel, respectively; and τ_h is the motor torque on the flywheel.

For non-sliding constraints, the ground reaction force (f_x, f_z) should remain inside a friction cone with a friction coefficient μ:

$$-\mu < \frac{f_x}{f_z} = \frac{x}{z_0} - \frac{\tau_h}{mgz_0} < \mu \tag{A.7}$$

which leads to the following constraint on x:

$$-\mu z_0 + \frac{\tau_h}{mg} < x < \mu z_0 + \frac{\tau_h}{mg}. \tag{A.8}$$

LIPPFM allows the robot to move one step to make the support polygon include x. However, here we need to find allowable motions which satisfy current foot contact constraints; so additionally, the CoM should be above the support polygon, which provides bounds x_{min} and x_{max} for x. Finally, x is constrained by

$$\max(-\mu z_0 + \frac{\tau_h}{mg},\ x_{min}) < x < \min(\mu z_0 + \frac{\tau_h}{mg},\ x_{max}). \tag{A.9}$$

From LIPPFM, the bounds on velocity $\dot{x}$ can be obtained. For example, with a step change in the rotational position of the flywheel of $\Delta\theta_b$, $\dot{x}$ is constrained by

$$\sqrt{\frac{g}{z_0}}(\underline{x} + \frac{J}{mz_0}\Delta\theta_{bmin}) < \dot{x} < \sqrt{\frac{g}{z_0}}(\bar{x} + \frac{J}{mz_0}\Delta\theta_{bmax}) \tag{A.10}$$

where $\underline{x}$ and $\bar{x}$ denote the lower and upper bounds of x.

With the aforementioned relations, an order of the magnitude of the maximum allowable motion velocity can be obtained. For example, consider a virtual character with a mass of $m = 79kg$; an inertia of $J = 3.125kgm^2$; the CoM height of $z_0 = 0.9m$; the minimum and maximum flywheel angles of $\Delta\theta_{bmin} = -\frac{1}{4}\pi$ and $\Delta\theta_{bmax} = \frac{3}{4}\pi$, respectively; the minimum and maximum CoM horizontal positions of $x_{min} = -0.03m$ and $x_{max} = 0.06m$, respectively; and the minimum and maximum hip torques of $\tau_{hmin} = -100Nm$ and $\tau_{hmax} = 100Nm$, respectively. The friction coefficient between its foot and the ground is $\mu = 1$. The gravitational acceleration is $g = 9.81ms^{-2}$. According to (A.9), the lower and upper bounds of x are $\underline{x} = -0.03m$ and $\bar{x} = 0.06m$. Then according to (A.10), the lower and upper bounds of the velocity $\dot{x}$ are $\underline{\dot{x}} = -0.2ms^{-1}$ and $\bar{\dot{x}} = 0.5ms^{-1}$, respectively.

Appendix B

Representation of Our Motion Capture System

We use the Advanced Real-time Tracking [Adv 2007] as our motion capture system. This system requires the following equipments:

- a set of markers,
- four tracking cameras,
- room calibration tools,
- a computer for data processing.

Markers are attached on the operator's body (see Fig.B.1). Each marker is spherical

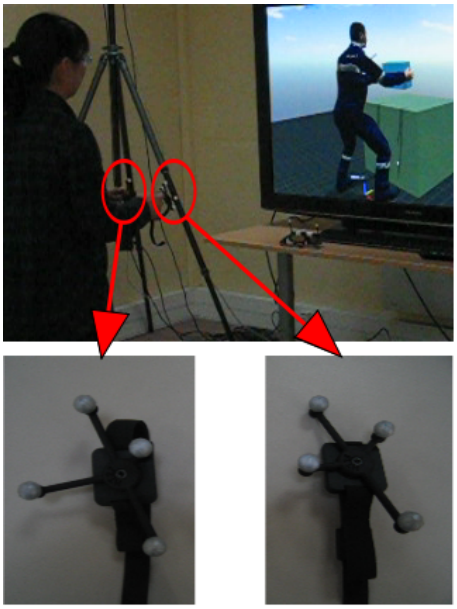


Figure B.1: Markers are attached to the hands of the operator.

shaped, covered with retro-reflective material. Each tracked body is associated with a

group markers geometrically configured into a certain form. The configurations of marker groups are different from each other, so that the system can distinguish one marker from another.

The motions of the markers are scanned by four cameras fixed at four corners of the room (see Fig.B.2). The positions of the cameras should be chosen so that the markers

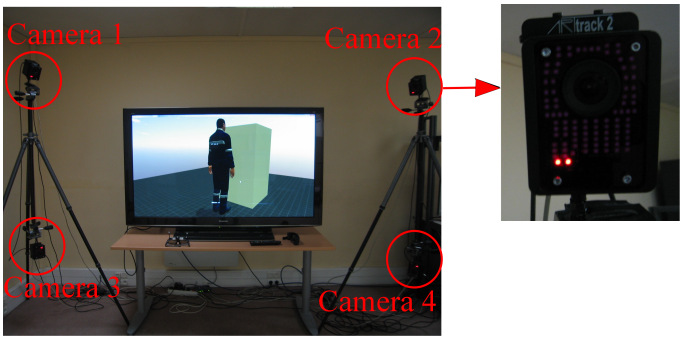


Figure B.2: Four cameras are used to scan the motions of the markers.

can be seen by as many cameras as possible. The motion tracking data scanned by the four cameras are sent to a computer for data processing.

Before starting motion capture, the groups of markers assigned to different tracked bodies are calibrated, so that the system will be able to recognize each of them during motion capture. Moreover, the room where motion capture takes place is also calibrated each time a camera's position changes, by using the room calibration tools. The calibration tool defines the origin and the axes of the coordinate system, in which the position and orientation of a tracked body is expressed.

This is a 6-DoF tracking system; so finally we can obtain both the positions and the orientations of the tracked bodies.

Bibliography

[Abe 2007] Yeuhi Abe, Marco da Silva and Jovan Popović. *Multiobjective control with frictional contacts.* In Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 249–258, 2007.

[Adv 2007] Advanced Real-time Tracking GmbH. *ARTtrack & DTrack User Manual*, 2007.

[Birk 1997] H. Birk, T.B. Moeslund and C.B. Madsen. *Real-Time Recognition of Hand Alphabet Gestures Using Principal Component Analysis.* In 10th Scandinavian conference on image analysis, pages 261–268, June 1997.

[Boulic 1996] R. Boulic, R. Mas and D. Thalmann. *A robust approach for the control of the center of mass with inverse kinetics.* Computers & Graphics, vol. 20, no. 5, pages 693–701, 1996.

[Bouyarmane 2010] K. Bouyarmane and A. Kheddar. *Static multi-contact inverse problem for multiple humanoid robots and manipulated objects.* In Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on, pages 8–13, dec. 2010.

[Bretl 2008] T. Bretl and S. Lall. *Testing Static Equilibrium for Legged Robots.* IEEE Transactions on Robotics, vol. 24, no. 4, pages 794–807, aug. 2008.

[C. Lawrence 1997] J. L. Zhou C. Lawrence and A. Tits. *User's Guide for CFSQP: A C Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality Constraints, version 2.5.* Institute for Systems Research, University of Maryland, College Park, MD, 1997.

[Calinon 2009] S. Calinon, P. Evrard, E. Gribovskaya, A. Billard and A. Kheddar. *Learning collaborative manipulation tasks by demonstration using a haptic interface.* In Advanced Robotics, 2009. ICAR 2009. International Conference on, pages 1–6, june 2009.

[Callennec 2006] Benoît Le Callennec and Ronan Boulic. *Interactive motion deformation with prioritized constraints.* Graphical Models, vol. 68, no. 2, pages 175–193, 2006.

[Chipalkatty 2010] R. Chipalkatty and M. Egerstedt. *Human-in-the-Loop: Terminal constraint receding horizon control with human inputs.* In 2010 IEEE International Conference on Robotics and Automation (ICRA), pages 2712–2717, may 2010.

[Colgate 1989] E. Colgate and N. Hogan. *The interaction of robots with passive environments: Application to force feedback control.* In Fourth International Conference on Advanced Robotics, 1989.

[Collette 2007] C. Collette, A. Micaelli, C. Andriot and P. Lemerle. *Dynamic balance control of humanoids for multiple grasps and non coplanar frictional contacts.* In 7th IEEE-RAS International Conference on Humanoid Robots, pages 81–88, 2007.

[Collette 2008] C. Collette, A. Micaelli, C. Andriot and P. Lemerle. *Robust balance optimization control of humanoid robots with multiple non coplanar grasps and frictional contacts.* In IEEE International Conference on Robotics and Automation (ICRA), pages 3187–3193, 2008.

[Cooper 2007] Seth Cooper, Aaron Hertzmann and Zoran Popović. *Active learning for real-time motion controllers.* In ACM SIGGRAPH 2007 papers, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[Coros 2010] Stelian Coros, Philippe Beaudoin and Michiel van de Panne. *Generalized biped walking control.* ACM Trans. Graph., vol. 29, pages 130:1–130:9, July 2010.

[Da Silva 2008a] M. Da Silva, Y. Abe and J. Popović. *Simulation of Human Motion Data using Short-Horizon Model-Predictive Control.* Computer Graphics Forum, vol. 27, no. 2, pages 371–380, 2008.

[da Silva 2008b] Marco da Silva, Yeuhi Abe and Jovan Popović. *Interactive simulation of stylized human locomotion.* In ACM SIGGRAPH 2008 papers, SIGGRAPH '08, pages 82:1–82:10, New York, NY, USA, 2008. ACM.

[de Schutter 2005] J. de Schutter, J. Rutgeerts, E. Aertbelien, F. de Groote, T. de Laet, T. Lefebvre, W. Verdonck and H. Bruyninckx. *Unified Constraint-Based Task Specification for Complex Sensor-Based Robot Systems.* In IEEE International Conference on Robotics and Automation (ICRA), pages 3607–3612, april 2005.

[De Schutter 2007] Joris De Schutter, Tinne De Laet, Johan Rutgeerts, Wilm Decré, Ruben Smits, Erwin Aertbeliën, Kasper Claes and Herman Bruyninckx.

Constraint-based Task Specification and Estimation for Sensor-Based Robot Systems in the Presence of Geometric Uncertainty. The International Journal of Robotics Research, vol. 26, no. 5, pages 433–455, 2007.

[Demircan 2008] Emel Demircan, Luis Sentis, Vincent Sapio and Oussama Khatib. *Human Motion Reconstruction by Direct Control of Marker Trajectories.* In Advances in Robot Kinematics: Analysis and Design, pages 263–272. Springer Netherlands, 2008.

[Ekvall 2007] S. Ekvall and D. Kragic. *Learning and Evaluation of the Approach Vector for Automatic Grasp Generation and Planning.* In Robotics and Automation, 2007 IEEE International Conference on, pages 4715–4720, april 2007.

[Escande 2010] A. Escande, N. Mansard and P.-B. Wieber. *Fast resolution of hierarchized inverse kinematics with inequality constraints.* In IEEE International Conference on Robotics and Automation (ICRA), pages 3733–3738, may 2010.

[Faloutsos 2001] Petros Faloutsos, Michiel van de Panne and Demetri Terzopoulos. *Composable controllers for physics-based character animation.* In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, SIGGRAPH '01, pages 251–260, New York, NY, USA, 2001. ACM.

[Fang 2003] Anthony C. Fang and Nancy S. Pollard. *Efficient synthesis of physically valid human motion.* ACM Trans. Graph., vol. 22, no. 3, pages 417–426, July 2003.

[Fasse 1998] Ernest D. Fasse and Peter C. Breedveld. *Modeling of Elastically Coupled Bodies: Part I—General Theory and Geometric Potential Function Method.* Journal of Dynamic Systems, Measurement, and Control, vol. 120, no. 4, pages 496–500, 1998.

[G. A. ten Holt 2007] M. J. T. Reinders G. A. ten Holt and E. A. Hendriks. *Multidimensional dynamic time warping for gesture recognition.* In Annual Conference on the Advanced School for Computing and Imaging, 2007.

[Ginsberg] Carol M Ginsberg and Delle Maxwell. *Graphical Marionette.*

[Gleicher 1998] Michael Gleicher. *Retargetting motion to new characters.* In Proceedings of the 25th annual conference on Computer graphics and interactive techniques, SIGGRAPH '98, pages 33–42, New York, NY, USA, 1998. ACM.

[Gribovskaya 2011] E. Gribovskaya, S.M. Khansari-Zadeh and A. Billard. *Learning Non-linear Multivariate Dynamics of Motion in Robotic Manipulators.* Int. J. Rob. Res., vol. 30, no. 1, pages 80–117, January 2011.

[Guenter 2007] F. Guenter, M. Hersch, S. Calinon and A. Billard. *Reinforcement Learning for Imitating Constrained Reaching Movements.* Advanced Robotics, Special Issue on Imitative Robots, vol. 21, no. 13, pages 1521–1544, 2007.

[Hannaford 2002] B. Hannaford and Jee-Hwan Ryu. *Time-domain passivity control of haptic interfaces.* IEEE Transactions on Robotics and Automation, vol. 18, no. 1, pages 1–10, feb 2002.

[Harada 2004] K. Harada, H. Hirukawa, F. Kanehiro, K. Fujiwara, K. Kaneko, S. Kajita and M. Nakamura. *Dynamical balance of a humanoid robot grasping an environment.* In Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on, volume 2, pages 1167–1173, sept.-oct. 2004.

[Harada 2006] K. Harada, K. Hauser, T. Bretl and J.-C. Latombe. *Natural Motion Generation for Humanoid Robots.* In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 833–839, oct. 2006.

[Hauser 2011] Kris Hauser and Victor Ng-Thow-Hing. *Randomized multi-modal motion planning for a humanoid robot manipulation task.* The International Journal of Robotics Research, vol. 30, no. 6, pages 678–698, 2011.

[Herdt 2010] Andrei Herdt, Holger Diedam, Pierre-Brice Wieber, Dimitar Dimitrov, Katja Mombaur and Moritz Diehl. *Online Walking Motion Generation with Automatic Foot Step Placement.* Advanced Robotics -Utrecht-, vol. 24, no. 5-6, pages 719–737, 2010.

[Jain 2009a] Sumit Jain and C. Karen Liu. *Interactive synthesis of human-object interaction.* In Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '09, pages 47–53, 2009.

[Jain 2009b] Sumit Jain, Yuting Ye and C. Karen Liu. *Optimization-Based Interactive Motion Synthesis.* ACM Transaction on Graphics, vol. 28, no. 1, pages 1–10, 2009.

[Kajita 1991] S. Kajita and K. Tani. *Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode.* In Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on, pages 1405 –1411 vol.2, apr 1991.

[Kallmann 1998] Marcelo Kallmann and Daniel Thalmann. *Modeling Objects for Interaction Tasks.* In Proc. Eurographics Workshop on Animation and Simulation, pages 73–86, 1998.

[Kanoun 2009] Oussama Kanoun, Florent Lamiraux, Pierre-Brice Wieber, Fumio Kanehiro, Eiichi Yoshida and Jean-Paul Laumond. *Prioritizing linear equality and inequality systems: Application to local motion planning for redundant robots.* In IEEE International Conference on Robotics and Automation (2009), pages 2939–2944, may 2009.

[Kanoun 2011a] O. Kanoun, F. Lamiraux and P.-B. Wieber. *Kinematic Control of Redundant Manipulators: Generalizing the Task-Priority Framework to Inequality Task.* IEEE Transactions on Robotics, vol. 27, no. 4, pages 785–792, aug. 2011.

[Kanoun 2011b] Oussama Kanoun, Jean-Paul Laumond and Eiichi Yoshida. *Planning foot placements for a humanoid robot: A problem of inverse kinematics.* The International Journal of Robotics Research, vol. 30, no. 4, pages 476–485, 2011.

[Khatib 1986] Oussama Khatib. *Real-Time Obstacle Avoidance for Manipulators and Mobile Robots.* The International Journal of Robotics Research, vol. 5, no. 1, pages 90–98, 1986.

[Khatib 2004] Oussama Khatib, Luis Sentis, Jaeheung Park and James Warren. *Whole-Body Dynamic Behavior and Control of Human-like Robots.* I. J. Humanoid Robotics, vol. 1, no. 1, pages 29–43, 2004.

[Khatib 2008] Oussama Khatib, Luis Sentis and Jae-Heung Park. *A Unified Framework for Whole-Body Humanoid Robot Control with Multiple Constraints and Contacts.* In European Robotics Symposium 2008, volume 44 of *Springer Tracts in Advanced Robotics*, pages 303–312. Springer Berlin / Heidelberg, 2008.

[Kim 2010] Jong-Phil Kim and Jeha Ryu. *Robustly Stable Haptic Interaction Control using an Energy-bounding Algorithm.* The International Journal of Robotics Research, vol. 29, no. 6, pages 666–679, 2010.

[Komura 2004] Taku Komura, Howard Leung and James Kuffner. *Animating reactive motions for biped locomotion.* In Proceedings of the ACM symposium on Virtual reality software and technology, VRST '04, pages 32–40, New York, NY, USA, 2004. ACM.

[Komura 2005] T. Komura, A. Nagano, H. Leung and Y. Shinagawa. *Simulating pathological gait using the enhanced linear inverted pendulum model.* Biomedical Engineering, IEEE Transactions on, vol. 52, no. 9, pages 1502–1513, sept. 2005.

[Krüger 2008] J. Krüger and D. Surdilovic. *Robust control of force-coupled human-robot-interaction in assembly processes.* CIRP Annals - Manufacturing Technology, vol. 57, no. 1, pages 41–44, 2008.

[Lee 2009] Sung-Hee Lee and Ambarish Goswami. *The Reaction Mass Pendulum (RMP) Model for Humanoid Robot Gait and Balance Control.* In Ben Choi, editeur, Humanoid Robots. InTech, 2009.

[Lee 2010] Dongheui Lee, Christian Ott and Yoshihiko Nakamura. *Mimetic Communication Model with Compliant Physical Contact in Human-Humanoid Interaction.* The International Journal of Robotics Research, vol. 29, no. 13, pages 1684–1704, 2010.

[Lee 2011] Jaemin Lee, N. Mansard and Jaeheung Park. *Intermediate desired value approach for continuous transition among multiple tasks of robots.* In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 1276–1282, may 2011.

[Liégeois 1977] A. Liégeois. *Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms.* IEEE Transactions on Systems, Man and Cybernetics, vol. 7, no. 12, pages 868–871, dec. 1977.

[Liu 2002] C. Karen Liu and Zoran Popović. *Synthesis of complex dynamic character motion from simple animations.* In Proceedings of the 29th annual conference on

Computer graphics and interactive techniques, SIGGRAPH '02, pages 408–416, New York, NY, USA, 2002. ACM.

[Liu 2011a] Mingxing Liu, Alain Micaelli, Paul Evrard, Adrien Escande and Claude Andriot. *An energy based two level prioritized control for virtual humans.* In IEEE International Conference on Robotics and Biomimetics (ROBIO), pages 186–191, dec. 2011.

[Liu 2011b] Mingxing Liu, Alain Micaelli, Paul Evrard, Adrien Escande and Claude Andriot. *Interactive dynamics and balance of a virtual character during manipulation tasks.* In IEEE International Conference on Robotics and Automation (ICRA), pages 1676–1682, may 2011.

[Liu 2012a] M. Liu, A. Micaelli, P. Evrard, A. Escande and C. Andriot. *Interactive Virtual Humans: A Two-Level Prioritized Control Framework With Wrench Bounds.* IEEE Transactions on Robotics, vol. PP, no. 99, pages 1–14, 2012.

[Liu 2012b] Mingxing Liu, Alain Micaelli, Paul Evrard and Adrien Escande. *Task-driven Posture Optimization for Virtual Characters.* In Eurographics/ ACM SIGGRAPH Symposium on Computer Animation, pages 155–164, 2012.

[Ma 2009] Liang Ma, Wei Zhang, Damien Chablat, Fouad Bennis and François Guillaume. *Multi-objective optimisation method for posture prediction and analysis with consideration of fatigue effect and its application case.* Computers & Industrial Engineering, vol. 57, no. 4, pages 1235–1246, November 2009.

[Mansard 2009] N. Mansard, O. Khatib and A. Kheddar. *A Unified Approach to Integrate Unilateral Constraints in the Stack of Tasks.* IEEE Transactions on Robotics, vol. 25, no. 3, pages 670–685, june 2009.

[Mansour 2011] Darine Mansour, Alain Micaelli and Pierre Lemerle. *A computational approach for push recovery in case of multiple noncoplanar contacts.* In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pages 3213–3220, sept. 2011.

[Miller 2003] A.T. Miller, S. Knoop, H.I. Christensen and P.K. Allen. *Automatic grasp planning using shape primitives.* In Robotics and Automation, 2003. Proceedings.

ICRA '03. IEEE International Conference on, volume 2, pages 1824–1829, sept. 2003.

[Monzani 2000] J. S. Monzani, P. Baerlocher, R. Boulic and D. Thalmann. *Using an intermediate skeleton and inverse kinematics for motion retargeting.* In Proc. Eurographics 2000, volume 19, pages 11–19, 2000. Comput. Graphics Lab., Swiss Fed. Inst. of Technol., Lausanne, Switzerland.

[Muico 2009] Uldarico Muico, Yongjoon Lee, Jovan Popović and Zoran Popović. *Contact-aware nonlinear control of dynamic characters.* ACM Trans. Graph., vol. 28, pages 81:1–81:9, July 2009.

[Müller 2005] Meinard Müller, Tido Röder and Michael Clausen. *Efficient content-based retrieval of motion capture data.* ACM Trans. Graph., vol. 24, no. 3, pages 677–685, July 2005.

[Multon 2009] Franck Multon, Richard Kulpa, Ludovic Hoyet and Taku Komura. *Interactive animation of virtual humans based on motion capture data.* Computer Animation and Virtual Worlds, vol. 20, no. 5-6, pages 491–500, 2009.

[Nguyen 2010] Nam Nguyen, Nkenge Wheatland, David Brown, Brian Parise, C. Karen Liu and Victor Zordan. *Performance capture with physical interaction.* In Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '10, pages 189–195, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics Association.

[Ott 2008] Christian Ott, Dongheui Lee and Yoshihiko Nakamura. *Motion Capture based Human Motion Recognition and Imitation by Direct Marker Control.* In Proceedings of the 2008 IEEE International Conference on Humanoid Robots, pages 399–405, 2008.

[Pastor 2011] P. Pastor, M. Kalakrishnan, S. Chitta, E. Theodorou and S. Schaal. *Skill learning and task outcome prediction for manipulation.* In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 3828–3834, may 2011.

[Peinado 2009] M. Peinado, D. Meziat, D. Maupu, D. Raunhardt, D. Thalmann and R. Boulic. *Full-Body Avatar Control with Environment Awareness.* IEEE Computer Graphics and Applications, vol. 29, no. 3, pages 62–75, may-june 2009.

[Pratt 1996] J. Pratt, A. Torres, P. Dilworth and G. Pratt. *Virtual actuator control.* In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), volume 3, pages 1219–1226, nov 1996.

[Pratt 2006] J. Pratt, J. Carff, S. Drakunov and A. Goswami. *Capture Point: A Step toward Humanoid Push Recovery.* In Humanoid Robots, 2006 6th IEEE-RAS International Conference on, pages 200 –207, dec. 2006.

[Rabiner 1989] L.R. Rabiner. *A tutorial on hidden Markov models and selected applications in speech recognition.* Proceedings of the IEEE, vol. 77, no. 2, pages 257–286, feb 1989.

[Raunhardt 2011] Daniel Raunhardt and Ronan Boulic. *Immersive singularity-free full-body interactions with reduced marker set.* Computer Animation and Virtual Worlds, 2011.

[Rennuit 2005] A. Rennuit, A. Micaelli, X. Merlhiot, C. Andriot, F. Guillaume, N. Chevassus, D. Chablat and P. Chedmail. *Passive control architecture for virtual humans.* In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1432–1437, aug. 2005.

[Rennuit 2006] A. Rennuit. *Contribution au Contrôle des Humains Virtuels Interactifs.* Ph.D. thesis, Ecole Centrale de Nantes, pages 73–77, 2006. in French.

[Rigoll 1998] Gerhard Rigoll, Andreas Kosmala and Stefan Eickeler. *High performance real-time gesture recognition using Hidden Markov Models.* In Ipke Wachsmuth and Martin Fröhlich, editeurs, Gesture and Sign Language in Human-Computer Interaction, volume 1371 of *Lecture Notes in Computer Science*, pages 69–80. Springer Berlin / Heidelberg, 1998. 10.1007/BFb0052990.

[Saab 2009] L. Saab, P. Soueres and J.Y. Fourquet. *Coupling manipulation and locomotion tasks for a humanoid robot.* In International Conference on Advances in Computational Tools for Engineering Applications (ACTEA), pages 84–89, july 2009.

[Saab 2011] L. Saab, N. Mansard, F. Keith, J-Y. Fourquet and P. Soueres. *Generation of dynamic motion for anthropomorphic systems under prioritized equality and in-*

equality constraints. In IEEE International Conference on Robotics and Automation (ICRA), pages 1091–1096, may 2011.

[Salini 2011] J. Salini, V. Padois and P. Bidaud. *Synthesis of complex humanoid whole-body behavior: A focus on sequencing and tasks transitions.* In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 1283 –1290, may 2011.

[Salini 2012] J. Salini. *Dynamic control for the task/posture coordination of humanoids: toward synthesis of complex activities.* Ph.D. thesis, University of Pierre and Marie Curie, 2012.

[Sentis 2004] L. Sentis and O. Khatib. *Task-Oriented Control of Humanoid Robots Through Prioritization.* In IEEE RAS/RSJ International Conference on Humanoid Robots, 2004.

[Sentis 2005] L. Sentis and O. Khatib. *Control of Free-Floating Humanoid Robots Through Task Prioritization.* In IEEE International Conference on Robotics and Automation (ICRA), pages 1718–1723, april 2005.

[Sentis 2010] L. Sentis, Jaeheung Park and O. Khatib. *Compliant Control of Multi-Contact and Center of Mass Behaviors in Humanoid Robots.* IEEE Transactions on Robotics, vol. 26, no. 3, pages 483–501, june 2010.

[Shapiro 2008] Ari Shapiro, Marcelo Kallmann and Petros Faloutsos. *Interactive motion correction and object manipulation.* In ACM SIGGRAPH, SIGGRAPH '08, pages 57:1–57:8, 2008.

[Siciliano 1991] B. Siciliano and J.-J.E. Slotine. *A general framework for managing multiple tasks in highly redundant robotic systems.* In Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on, pages 1211 –1216 vol.2, june 1991.

[Smits 2010] R. Smits. *Design of a Constraint-Based Methodology and Software Support.* Ph.D. thesis, Katholieke Universiteit Leuven, 2010.

[Spong 2005] M.W. Spong and F. Bullo. *Controlled symmetries and passive walking.* IEEE Transactions on Automatic Control, vol. 50, no. 7, pages 1025–1031, july 2005.

[Stasse 2008] O. Stasse, A. Escande, N. Mansard, S. Miossec, P. Evrard and A. Kheddar. *Real-time (self)-collision avoidance task on a hrp-2 humanoid robot.* In IEEE International Conference on Robotics and Automation (ICRA), pages 3200–3205, may 2008.

[Stephens 2010] B.J. Stephens and C.G. Atkeson. *Dynamic Balance Force Control for compliant humanoid robots.* In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1248–1255, oct. 2010.

[Stilman 2010] M. Stilman. *Global Manipulation Planning in Robot Joint Space With Task Constraints.* IEEE Transactions on Robotics, vol. 26, no. 3, pages 576–584, june 2010.

[Stulp 2010] F. Stulp, J. Buchli, E. Theodorou and S. Schaal. *reinforcement learning of full-body humanoid motor skills.* In humanoid robots (humanoids), 2010 10th ieee-ras international conference on, pages 405–410, 2010.

[Sung 2004] H. Sung. *Gaussian mixture regression and classification.* Ph.D. thesis, Rice University, 2004.

[Theodorou 2010] E. Theodorou, J. Buchli and S. Schaal. *a generalized path integral control approach to reinforcement learning.* no. 11, pages 3137–3181, 2010.

[Wiley 1997] Douglas J. Wiley and James K. Hahn. *Interpolation Synthesis of Articulated Figure Motion.* IEEE Comput. Graph. Appl., vol. 17, no. 6, pages 39–45, November 1997.

[Wu 2010] Jia-chi Wu and Zoran Popović. *Terrain-adaptive bipedal locomotion control.* ACM Trans. Graph., vol. 29, pages 72:1–72:10, July 2010.

[XDE] *XDE eXtended Dynamic Engine.* `http://www.kalisteo.fr/lsi/en/aucune/a-propos-de-xde`.

[Xinjilefu 2009] X. Xinjilefu, V. Hayward and H. Michalska. *Stabilization of the spatial double inverted pendulum using stochastic programming seen as a model of standing postural control.* In Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on, pages 367–372, dec. 2009.

[Yamane 2004] Katsu Yamane, James J. Kuffner and Jessica K. Hodgins. *Synthesizing animations of human manipulation tasks.* ACM Trans. Graph., vol. 23, no. 3, pages 532–539, August 2004.

[Yang 2004] J. Yang, R. Timothy Marler, H. Kim, Jasbir S. Arora and K. Abdel-Malek. *Multi-objective Optimization for Upper Body Posture Prediction.* In 10th AIAA/ISSMO multidisciplinary analysis and optimization conference, 2004.

[Ye 2010] Yuting Ye and C. Karen Liu. *Optimal feedback control for character animation using an abstract model.* ACM Trans. Graph., vol. 29, pages 74:1–74:9, July 2010.

[Yoshida 2006] E. Yoshida, O. Kanoun, C. Esteves and J.-P. Laumond. *Task-driven Support Polygon Reshaping for Humanoids.* In 6th IEEE-RAS International Conference on Humanoid Robots, pages 208–213, dec. 2006.

[Yoshida 2010] Eiichi Yoshida, Mathieu Poirier, Jean-Paul Laumond, Oussama Kanoun, Florent Lamiraux, Rachid Alami and Kazuhito Yokoi. *Pivoting based manipulation by a humanoid robot.* Autonomous Robots, vol. 28, pages 77–88, 2010.

[Zordan 2002] Victor Brian Zordan and Jessica K. Hodgins. *Motion capture-driven simulations that hit and react.* In Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation, SCA '02, pages 89–96, 2002.

[Zordan 2005] Victor Brian Zordan, Anna Majkowska, Bill Chiu and Matthew Fast. *Dynamic response for motion capture animation.* ACM Trans. Graph., vol. 24, no. 3, pages 697–701, July 2005.

List of Author's Publications

Journal Publications

[1] M. Liu, A. Micaelli, P. Evrard, A. Escande and C. Andriot. *Interactive Virtual Humans: A Two-Level Prioritized Control Framework With Wrench Bounds.* IEEE Transactions on Robotics, vol. PP, no. 99, pages 1–14, 2012.

Conference Publications

[1] Mingxing Liu, Alain Micaelli, Paul Evrard and Adrien Escande. *Task-driven Posture Optimization for Virtual Characters.* In Eurographics/ ACM SIGGRAPH Symposium on Computer Animation, pages 155–164, 2012.

[2] Mingxing Liu, Alain Micaelli, Paul Evrard, Adrien Escande, and Claude Andriot. An energy based two level prioritized control for virtual humans. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 186–191, dec. 2011.

[3] Mingxing Liu, Alain Micaelli, Paul Evrard, Adrien Escande, and Claude Andriot. Interactive dynamics and balance of a virtual character during manipulation tasks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1676–1682, may 2011.